Méthodes de carte auto-organisatrice par mélange de lois contraintes. Application à l'exploration dans les tableaux de contingence textuels

Rodolphe Priam

N° d'ordre: 2879

THÈSE

présentée

DEVANT L'UNIVERSITÉ DE RENNES 1

pour obtenir

le grade de : ***DOCTEUR DE L'UNIVERSITE DE RENNES 1***
Mention : Informatique

par

Rodolphe Priam

Équipe d'accueil : TEXMEX (IRISA, RENNES)

École Doctorale : Mathématiques, Informatique, Signal,
Electronique et Télécommunication

Composante universitaire : Ifsic

Titre de la thèse :

Méthodes de carte auto-organisatrice par mélange de lois contraintes. Application à l'exploration dans les tableaux de contingence textuels.

soutenue le 17 Octobre 2003 devant

COMPOSITION DU JURY

Gérard	Govaert	Professeur à l'Université de Compiègne	Rapporteur
Yves	LeChevallier	Directeur de Recherche INRIA	Rapporteur
Ludovic	Lebart	Directeur de Recherche CNRS	Examinateur
Israël-César	Lerman	Professeur à l'IFSIC	Président
Djamel Abdelkader	Zighed	Professeur à l'Université de Lyon 2	Examinateur
Annie	Morin	Maître de Conférences à l'IFSIC	Directrice

À mes parents
À ma soeur jumelle
À Cécile

Remerciements

Je remercie M. Israël-César Lerman qui me fait l'honneur de présider ce jury, M. Gérard Govaert et M. Yves LeChevallier qui me font l'honneur d'accepter la charge de Rapporteur, M. Ludovic Lebart et M. Djamel Abdelkader Zighed qui me font l'honneur de juger ce travail en tant qu'Examinateur. Je tiens tout particulièrement à remercier l'ensemble des membres du jury pour leurs remarques qui ont permis d'éclairer des points essentiels du document.

Je remercie le directeur de l'IRISA, M. Claude Labit, de m'avoir accepté dans son laboratoire, ainsi que les directeurs des projets SIGMA2, M. François Legland puis TEXMEX, M. Patrick Gros, de m'avoir permis de faire ces recherches au sein de leur équipe.

Je remercie l'ensemble des personnes qui ont contribué de près ou de loin à ces recherches, les anciens collocataires de bureau et plus particulièrement Frédéric Cérou pour sa gentillesse, les membres de l'équipe METISS, dont Frédéric Bimbot, Raphaël Blouet, Mouhamadou Seck, Matthieu Ben et Michael Betser pour leur bonne humeur.

Je remercie ma Directrice de Thèse, Mme Annie Morin, pour son soutien, son aide, et ses conseils toujours avisés.

Merci à tous les autres.

Notations principales

On note les variables aléatoires par des majuscules et leur réalisation par des minuscules, et en gras les suite de ces v.a. ainsi que leurs réalisations.

$\mathcal{D} = (x_1, x_2, \cdots, x_I)$ ou $\mathbf{x}$	corpus : échantillon observé, réalisation de $\mathbf{X}$
$\mathcal{V} = (v_1, v_2, \cdots, v_j, \cdots, v_J)$	vocabulaire : label des variables ou mots
I	taille de l'espace des individus
$\mathcal{I} = \{1, 2, \cdots, I\}$	ensemble des indices de documents
J	taille de l'espace des variables
$\mathcal{J} = \{1, 2, \cdots, J\}$	ensemble des indices de mots
X_i ou X	variable aléatoire d'un individu document
x_i	document observé, réalisation de X_i ou X
$\mathbf{X} = (X_1, X_2, \cdots, X_i, \cdots, X_I)$	suite de v.a. X_i observée ou corpus aléatoire
D	matrice du corpus
N	matrice de comptage
B	matrice binaire
$f_{j\|i}$	probabilité conditionnelle empirique
$f_{i\bullet}$	vecteur de distribution conditionnelle empirique
$N_{i\bullet}$	taille du document x_i
f_i	probabilité empirique du document x_i
$N_{\bullet j}$	total des occurrences du mot v_j dans le corpus
f_j	probabilité empirique du mot v_j
D_I	matrice de poids des documents
D_J	matrice de poids des mots
Z_i ou Z	variable aléatoire latente non observée
z_i	réalisation de Z_i
$\mathbf{Z} = (Z_i)_i$	variables non observées
$\mathbf{z} = (z_i)_i$	réalisation de $\mathbf{Z}$
$\mathcal{K} = \{1, 2, \cdots, K\}$	espace des valeurs prises par Z discrète
m_k	centre J-dimensionnel de classe
$\mathbf{m} = (m_k)_k$	ensemble des centres de classe

θ	vecteur de paramètres
$P_{j\|k}$	probabilité du mot v_j dans la classe k-ième
$P_{k\bullet}$	vecteur de probabilités de composante $P_{j\|k}$
π_k	coefficient mélangeant du k-ième facteur
C_k	classe k-ième dans une partition de $\mathcal{D}$
$k \sim l$	k voisin de l sur un graphe
$\mathcal{N}_k$	ensemble des voisins de k sur un graphe
h	fonction de voisinage établie sur un graphe
ξ_k	coordonnées sur le plan de C_k
$H = (h(\xi_k, \xi_l))_{kl}$	matrice des voisinages
μ_{ik}	coefficient d'appartenance floue à une classe
μ	matrice de classification floue
c_{ik}	coefficient binaire d'affectation aux classes $(C_k)_k$
$\mathbf{C}$	matrice de classification
$\delta(\bullet, \bullet)$	Delta de Kronecker, $\delta(a, b) = 1$ ssi $a = b$

Chapitre 1

Introduction

L'*Analyse des Données* forme le domaine des méthodes exploratoires qui permettent de révéler et montrer les principales liaisons statistiques présentes dans une base de données. Elle a pour objectif de donner une vision synthétique et interprétable de l'organisation naturelle de données. Dans ce cadre, le document présent s'intéresse à la projection 2-dimensionnelle de la structure ou distribution au sens probabiliste du terme, de grands corpus de données par des méthodes exploratoires, issues non seulement de la statistique, mais aussi des modèles neuronaux. L'ensemble de ces méthodes se regroupe sous le terme *méthodes de cartographie*. **Ce document porte sur l'élaboration et l'étude de nouvelles méthodes de cartographie pour tableau de contingence. Pour ce faire, nous étudions la sous-classe des méthodes de Cartes Auto-Organisatrices ou Cartes de Kohonen et leur croisement avec les modèles probabilistes de mélange de lois en présentant une vue unifiée. La principale méthode développée s'apparente à l'Analyse Factorielle des Correspondances. Enfin, nous évaluons empiriquement les algorithmes proposés sur des données synthétiques et textuelles.**

La cartographie sur laquelle nous travaillons projette sur le plan un ensemble de vecteurs multidimensionnels de grande taille. Pour ce faire, elle optimise un critère choisi afin de préserver au mieux certaines caractéristiques globales ou locales de la distribution des données. Une cartographie fournit donc un résultat final sous la forme d'un paysage qui dessine une vue de l'*information* contenue dans un corpus de documents. Au delà de l'aspect visualisation d'un nuage de points multidimensionnels sur un plan, la cartographie offre de nombreuses autres applications possibles. **Les méthodes de cartographie étudiées effectuent une classification des données avec une contrainte de voisinage. C'est pourquoi, elles sont aptes à des applications en classification, et visualisation synthétique des données, ainsi qu'en navigation intuitive, indexation et en général à l'extraction automatique de connaissances.**

En analyse des données, se posent deux questions fondamentales lors de la construc-

tion ou l'utilisation des modèles : comment interpréter ou rendre interprétables les résultats. En effet, si les statistiques permettent généralement d'analyser et de tirer des conclusions précises d'un résultat, ce n'est pas toujours le cas des méthodes de réseaux de neurones artificiels souvent qualifiées de méthodes *boîte noire*. **Nous essayerons de répondre à la question d'interprétabilité des résultats, notamment dans le cas particulier de l'Analyse de Données Textuelles en posant des indicateurs numériques. Nous utiliserons essentiellement la distribution multinomiale pour sa simplicité et sa pertinence classificatoire reconnue dans le domaine**.

L'intérêt des données étudiées, **données textuelles**, réside dans la complexité du langage sous-jacent et de son caractère qualitatif et souvent approximatif : cela explique d'ailleurs le choix d'un formalisme statistique. Ce dernier est à même d'extraire les corrélations inconnues entre un grand nombre de variables tout en tenant compte du bruit, source d'erreur mais également de la diversité. On abordera le principe de réduction de dimension en général en le particularisant à la réduction contrainte en carte topologique. **Une façon originale de construire une cartographie à partir de la notion de segmentation d'image sera également développée et commentée.**

La donnée texte permettra d'appliquer nos résultats et de valider l'approche. Elle constitue l'essentiel de la suite de cette introduction, précédant la présentation du plan. Les notions de *document* et de *mot* sont en réalité très générales ; elles font référence dans la suite aux tableaux de contingence $I \times J$ quelconques pour des données appelées qualitatives ; par exemple ce sont les vecteurs de distributions discrètes finies comme on en trouve dans le domaine du traitement d'image, ou dans les sciences sociales. D'ailleurs, les tableaux de comptage englobent la plupart des lois connues pour peu que les échantillonnages empiriques soient convenables.

1.1 Traitements automatisés de corpus

Les domaines de recherche visant l'exploitation des données textuelles connaissent un essor important avec la croissance des capacités de stockage informatique. Le traitement automatique des corpus textuels est pourtant le sujet de recherches intensives depuis plusieurs décennies. Il se pose aujourd'hui en problématique majeure depuis quelques années avec le développement des moyens et usage des communications électroniques en tout lieu et tout temps : mails, newsgroups, Internet, télétravail, enseignement à distance, téléconférence... Exploiter les nombreuses bases de données structurées, hétérogènes ou multimédia, distribuées, dynamiques, et en volume croissant avec des thématiques apparaissant, disparaissant, se déplaçant, évoluant ou *mutant* sans cesse, reste encore aujourd'hui un véritable challenge scientifique et technologique. Le multimédia est l'information d'aujourd'hui et plus encore de demain ; son usage suppose la maîtrise du son, de l'image fixe et animée ainsi que du texte.

Même aujourd'hui où le multimédia se développe rapidement, la donnée texte demeure largement le media le plus employé ; justement, le document présent s'intéresse à la manipulation du média texte dans le but d'en obtenir de l'information pertinente et accessible, de façon rapide et efficace.

L'analyse textuelle peut se répartir en différentes thématiques, suivant l'objectif que l'on se fixe, elle s'intéresse à :

- La représentation des données textuelles, par exemple, une représentation sous forme vectorielle, une représentation sous forme séquentielle,
- L'indexation de corpus et leur interrogation afin de retrouver à partir d'une question les documents les plus pertinents,
- La classification qui construit des ensembles de textes le plus homogène possible,
- La catégorisation ou filtrage des textes qui placent les textes dans des catégories prédéfinies en essayant de cibler la "bonne" catégorie,
- La visualisation des textes, c'est-à-dire rechercher une représentation intuitive et synthétique pour observer les caractéristiques ou structure d'un corpus (mot, phrase, paragraphe, partie, texte),
- L'étude de contenu ou comment obtenir une information intelligible et compréhensible pour rechercher les relations entre les mots ou les textes du corpus,
- La navigation dans un corpus, problématique d'IHM (Interface Homme Machine),
- La détection d'un nouveau thème, ou comment détecter une nouvelle classe de textes ou de mots,
- La construction de dictionnaires, de synonymes, antonymes, définition, ou leur utilisation,
- La construction des modèles de langages ou de grammaire qui offre un moyen de trouver la probabilité d'apparition d'un mot à la suite de précédents, le moyen de donner le genre grammatical d'un mot dans une phrase, et consolider les modèles en reconnaissance de la parole, les correcteurs orthographiques et les traducteurs automatiques,
- La construction de réseaux conceptuels pour la formalisation de connaissances,
- La création de résumés automatiques, et remplissage de formulaires, etc...

Ceci constitue la plupart des traitements envisageables à partir d'un corpus de textes ; il est possible d'en ajouter beaucoup d'autres en prenant en compte les interactions avec les autres média que sont le son et l'image. Par rapport à ces traitements, on distingue à part le codage des textes, comme étant le résultat d'un passage du format électronique ou numérique à un format à préciser, apte à une modélisation mathématique pour obtenir un résultat donné. Ce codage rejoint la problématique de la représentation des données puisque suivant le cas, on aura besoin du texte dans son intégralité ou seulement d'une partie, sous la forme de comptages, dont la nature influe sur la manière de stocker les données. Le langage est une donnée complexe, il est effectivement bruité, incertain du fait que non seulement son écrit puisse être entaché d'erreurs (orthographe, faute grammaticale, ponctuation, typographie, ...), mais

en plus, du fait du domaine du subjectif (ou qualitatif), il existe plusieurs manières de dire la même chose ou de la comprendre suivant le contexte. D'ailleurs, la langue possède de nombreux effets de style littéraire. Le mot lui même peut porter à confusion : synonyme, polysémie, homonymie, par exemple, rendent un terme finalement hasardeux. Enfin, comme le vocabulaire est très large et spécialisé suivant le domaine, les algorithmes développés pour résoudre tel ou tel problème peuvent être pris à défaut par des mots inconnus ou pire, un changement de langue. L'idéal serait d'avoir une machine capable de comprendre le sens de l'écrit, mais une telle machine n'existe pas encore, si ce n'est dans le domaine du vivant. **Notre domaine d'application sera principalement l'analyse de données textuelles, i.e. la visualisation (avec indicateurs quantitatifs) de la structure d'un corpus pour l'extraction de connaissances textuelles. Nous ferons un bref détour sur les problématiques de classification et de façon moindre la recherche de documents.** La suite de cette introduction développe brièvement la problématique générale de l'analyse textuelle et des méthodes statistiques employées dans le domaine, puis expose le plan du document en décrivant les chapitres à venir sur la cartographie.

1.2 Texte, corpus, terme, et dictionnaire

Un mot (ou terme) est une suite de caractères appartenant à un ensemble de caractères définis. Les mots d'un texte sont séparés par des caractères appelés séparateurs. Un texte t_i est une suite de mots écrits dans un langage donné suivant des règles précises de vocabulaire, d'orthographe, et de grammaire. Un corpus est un ensemble de textes t_i pour i allant de 1 à I, chaque texte représenté formellement par la notation x_i. On utilise une représentation numérique adaptée à son traitement[1, 2, 3] automatique pour en extraire de l'information. L'ensemble (ou sous-ensemble) des mots distincts cités dans un corpus est nommé dictionnaire (ou vocabulaire).

1.3 Représentation textuelle à partir du dictionnaire

On dispose d'un corpus de I documents notés $\mathcal{D} = \{x_i\}_{i=1}^{i=I}$ où x_i est le i-ème document. On construit[1] un dictionnaire de J mots notés $\mathcal{V} = \{v_j\}_{j=1}^{j=J}$ où v_j est le j-ième mot ou terme[2]. On notera les ensembles d'indices $\mathcal{I} = [1, I]$ et $\mathcal{J} = [1, J]$.

Représentation séquentielle

Une fois déterminé le dictionnaire, un texte est une suite de mot pris dans le dictionnaire. En négligeant la ponctuation et les mots non retenus, on obtient l'événement "$x_{i1}, x_{i2}, \cdots, x_{i|x_i|}$" où $|x_i|$ correspond à la taille du texte, le nombre de mots le composant et $x_{it} \in \mathcal{V}$. La représentation séquentielle évalue la probabilité de l'événement d'un mot donné conditionnellement à la suite le précédant. On fait l'hypothèse que la

[1] cf. partie suivante (vectoriel), et chapitre *Applications*

[2] ordonné par ordre d'arrivée, lexical ou même fréquentiel

probabilité d'un mot à la position t dans le texte ne dépend que des h mots le précédant ou plus précisément $min(h,t)$ pour les h premiers mots du texte. On appelle historique les mots précédents x_t, que l'on note h_t. On écrit alors la probabilité du $t^{\text{ième}}$ mot dans un texte comme :

$$P(x_t|x_{t-1},\cdots,x_{t-h},x_{t-h-1},\cdots,x_1) = P(x_t|h_t) \tag{1.1}$$

On en déduit finalement la probabilité d'un document ou suite de mots, en particularisant h_t à un texte donné observé x_i en indiçant par i :

$$P(x_i) = \prod_{t=1}^{t=|x_i|} P(x_{it}|h_{i_t}) \tag{1.2}$$

On en dérive les modèles de langage[4] les plus usités pour k=1,2,3 avec v_j est un événement *mot* : unigramme pour h=0, bigramme pour h=1, trigramme pour h=2.

Exemple 1
Pour la séquence de mots "a b c d e" et un historique de taille 2, on a :

$$P(a,b,c,d,e) = P(a)P(b|\underbrace{a}_{h_2})P(c|\underbrace{a,b}_{h_3})P(d|\underbrace{b,c}_{h_4})P(e|\underbrace{c,d}_{h_5})$$

Cette approche permet également de définir les cooccurrences de mots de manière générale. Par exemple on compte le nombre de couples de mots consécutifs (v_{j1}, v_{j2}) dans le corpus. La définition peut s'étendre à des contextes plus larges en comptant dans des fenêtres d'un nombre fixe de mots par exemple, ou même à des paragraphes entiers.

Un autre modèle suppose la suite de mots ou ses étiquettes comme une réalisation observée conditionnellement à un processus séquentiel caché $z_i = (z_{i1}, z_{i2}, \cdots, z_{i|x_i|})$, lui même suivant une chaîne de Markov. On parle de *Hidden Markov Model* (HMM) ou Chaîne de Markov Cachée. Il est éventuellement possible d'appliquer un opérateur particulier F sur un mot observé, donnant son genre grammatical par exemple pour construire un modèle non plus du vocabulaire mais grammatical. Nous posons directement le mot v_j du vocabulaire, mais remplacer par $F(v_j)$ amènerait à une même formulation. La variable aléatoire du processus cachée non observé permet de modéliser l'historique de façon implicite. Un HMM pose finalement la loi jointe suivante :

$$P(x_i, z_i) = P(z_{i1})\, P(x_{i1}|z_{i1}) \prod_{t=2}^{t=|x_i|} P(z_{it}|z_{it-1})\, P(x_{it}|z_{it})$$

Les z_{it} sont les valeurs non observées du processus caché, à valeurs discrètes appelées états. On appelle $P(z_{it}|z_{it-1})$ probabilité de transition, $P(x_{it}|z_{it})$ probabilité d'émission ; x_{it} est le mot du document x_i ayant lieu à la t-ième position.

Exemple 2
Pour la phrase "a b c d e", on factorise $P(a, b, c, d, e, z_1, z_2, z_3, z_4, z_5)$ *par :*

$$P(a|z_1)P(b|z_2)P(c|z_3)P(d|z_4)P(e|z_5)P(z_1)\prod_{t=2}^{t=5} P(z_t|z_{t-1})$$

Ce genre de méthode cherche à estimer la dynamique du processus inconnu à partir de données labellées ou non. Ensuite, on peut calculer la probabilité d'une séquence donnée ou son processus caché le plus certain. Il existe[5] des algorithmes numériques performants pour résoudre le problème de marginalisation $P(x_i) = \sum_{z_i} P(x_i, z_i)$.

Exemple 3
En reprenant le HMM pour la phrase "a b c d e", en supposant 2 états binaires pour les variables cachées, on a

$$\begin{aligned} & P(a, b, c, d, e) \\ &= \textstyle\sum_{\mathbf{z}} P(a, b, c, d, e, z_1, z_2, z_3, z_4, z_5) \\ &= \textstyle\sum_{(z_1,z_2,z_3,z_4,z_5)\in\{0,1\}^5} P(a, b, c, d, e, z_1, z_2, z_3, z_4, z_5) \end{aligned}$$

Cet exemple montre la problématique de ce genre de modèle dont la taille des espaces croît exponentiellement avec le nombre des paramètres, et s'avère très vaste même pour un nombre d'états réduit. Le formalisme markovien entre dans le cadre plus général des modèles graphiques. Contrairement aux méthodes vectorielles, les méthodes séquentielles permettent de prédire le mot le plus probable à la suite d'une séquence de mots. Elles servent donc en désambiguïsation notamment et en traitement du langage naturel notamment en reconnaissance du langage parlé. En général, ces méthodes ne dépassent pas un historique supérieur à 3 étant donné la difficulté d'estimation[4] des probabilités de séquences qui sont apprises sur un corpus donné. Il a été proposé plus récemment des modèles à maximum d'entropie encore plus performants sur certains *benchmarks*. Ils sont à base de modèles de champ de Markov qui formalisent une dépendance plus complexe (treillis) sur les variables aléatoires. Toutes ces méthodes sont finalement très dépendantes du domaine couvert dont le vocabulaire doit être connu ainsi que son emploi courant (du point de vue contextuel). Elles nécessitent d'ailleurs des corpus annotés (labels connus) de taille importante pour espérer "apprendre" (ou estimer) les probabilités de cooccurrences contextuelles du langage. Des auteurs travaillent même directement sur les séquences de lettres, permettant d'accélérer notablement les algorithmes au prix d'une perte de l'interprétabilité directe des résultats.

Représentation vectorielle

L'article [6] est le premier[1] à faire référence à une représentation vectorielle des textes, représentation[7] la plus fréquente encore de nos jours. La représentation vectorielle fait correspondre à chacun des mots du vocabulaire la composante d'un vecteur J-dimensionnel.

La donnée textuelle se prête bien à la construction d'un tableau de comptages, noté D ou N dans la suite, puisque le discours se formule par un alignement de mots tirés d'un vocabulaire de taille finie.

Définition 1
On définit un tableau de contingence comme le tableau $I \times J$, de cellules $n_{ij} \in \mathbb{N}_+$:

n_{11}	n_{1j}	n_{1J}	
n_{i1}	n_{ij}	n_{iJ}	$n_{i\bullet}$
n_{I1}	n_{Ij}	n_{IJ}	
	$n_{\bullet j}$		$n_{\bullet\bullet}$

Dans un tableau de contingence, les marges ou sommes sur les lignes ou colonnes ont un sens puisque le tableau croise deux variables. Ainsi, $n_{i\bullet} = \sum_j^J n_{ij}$, $n_{\bullet j} = \sum_i^I n_{ij}$, $n_{\bullet\bullet} = \sum_i^I n_{i\bullet} = \sum_j^J n_{\bullet j} = \sum_i^I \sum_j^J n_{ij}$.

Il est possible de construire divers tableaux textuels de contingence à partir des données brutes du comptage des mots dans le texte : tableau d'occurrences mot*document ou version binaire, tableaux de cooccurrences mots*mots ou documents*documents (cas issu des comptages ou des binaires), tableau binaire des modalités éclatées (ou tableau disjonctif complet), tableau de Burt (carré par transposition). Dans la suite, nous nous intéressons exclusivement aux matrices textuelles suivantes (bien que des applications aux autres cas soient envisageables) : $N = [N_{ij}]_{ij}$[3] et $B = [B_{ij}]_{ij}$[4], où N est la matrice ayant pour cellule N_{ij}, le nombre d'occurrences du j-ième mot du corpus dans le i-ème texte du corpus, et B la matrice de cellule binaire B_{ij} qui indique par 1 si le j-ième mot du corpus est apparu dans le i-ème texte du corpus au moins une fois.

Exemple 4
Pour trois textes de séquences "a b c d e", "a c d c", "d d e f", en prenant pour vocabulaire $\{c, d, e\}$ construit par choix des mots les plus fréquents et ordonnés lexicalement, on a :

$$N = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \text{ et } B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

L'intérêt de travailler sur des matrices de comptages consiste en la possibilité d'étudier directement le tableau des données en terme de distribution discrète. En effet, plusieurs points de vue différents pour étudier un tel tableau se présentent naturellement.

---

[3] $N = [N_{ij}]_{ij}$ où $N_{ij} = \#\{x_{i_t} = v_j, i_t \in [1, |x_i|]\}$ pour x_i ici noté séquentiellement

[4] $B = [B_{ij}]_{ij}$ où $B_{ij} = \begin{cases} 1 & \text{si } N_{ij} > 0 \\ 0 & \textit{sinon} \end{cases}$

Définition 2
Un vecteur textuel de comptage brut x_i est :

$$\forall i \in \mathcal{I}\ x_i = \left\{ N_{ij}\ ;\ j \in \mathcal{J} \right\} \tag{1.3}$$

Définition 3
Un vecteur textuel vu comme une distribution discrète empirique est le profil ligne :

$$\forall i \in \mathcal{I}\ f_{i\bullet} = \left\{ f_{j|i} = \frac{N_{ij}}{N_{i\bullet}}\ ;\ j \in \mathcal{J} \right\} \tag{1.4}$$

Définition 4
Un vecteur mot vu comme une distribution discrète empirique est le profil colonne :

$$\forall j \in \mathcal{J}\ f_{\bullet j} = \left\{ f_{i|j} = \frac{N_{ij}}{N_{\bullet j}}\ ;\ i \in \mathcal{I} \right\} \tag{1.5}$$

Définition 5
Une distribution jointe de la variable mot de modalités $\mathcal{J}$ (ou $\mathcal{V}$) croisée avec la variable document de modalités $\mathcal{I}$ (ou $\mathcal{D}$) s'écrit :

$$F = \left\{ f_{ij} = \frac{N_{ij}}{N_{\bullet\bullet}}\ ;\ (i,j) \in \mathcal{I} \times \mathcal{J} \right\} \tag{1.6}$$

Définition 6
Les marges[5] *de F sont notées :*

$$\begin{array}{l} \left\{ f_j = \frac{N_{\bullet j}}{N_{\bullet\bullet}}\ ;\ j \in \mathcal{J} \right\} \\ \left\{ f_i = \frac{N_{i\bullet}}{N_{\bullet\bullet}}\ ;\ i \in \mathcal{I} \right\} \end{array} \tag{1.7}$$

Ainsi, chaque document est représenté par un vecteur $x_i = (x_{i1}, x_{i2}, \cdots, x_{iJ})$. Dans cette représentation vectorielle, x_i est bien une observation vectorielle J-dimensionnelle, alors que dans la représentation séquentielle précédente, il s'agissait d'une séquence observée de mots $x_i = (x_{i1}, x_{i2}, \cdots, x_{it}, \cdots, x_{i|x_i|})$ avec $x_{it} \in \mathcal{V}$; ici, x_{ij} est une fréquence relative $f_{j|i}$. Or, malgré la perte d'information sur l'ordre des mots, la représentation vectorielle donne des résultats acceptables (bien que perfectibles) tout en étant en général aussi bons, sinon meilleurs que les méthodes séquentielles. Les vecteurs x_i permettent de construire la matrice de comptage.

Exemple 5 *Pour trois textes de séquences "a b c d e", "a c d c", "d d e f", en prenant pour vocabulaire $\{c, d, e\}$ construit par choix des mots les plus fréquents et ordonnés lexicalement, on a pour le texte x_2, le profil ligne :*

$$f_{2\bullet} = (\frac{2}{3}\ \ \frac{1}{3}\ \ 0) \text{ puisque } N_{2\bullet} = 3$$

[5] Elles seront renommées en cas d'ambiguïté possible en raison d'indiçages différents sur une somme.

Il est éventuellement possible de rajouter aux vecteurs de nouvelles composantes, comme par exemples les fréquences de cooccurrences. La matrice obtenue au final a la particularité d'être très creuse[6] car généralement peu de mots sont partagés par deux textes différents. Elle est également a priori très grande car le vocabulaire (mots différents) du langage parlé croît assez rapidement avec le nombre de mots énoncés pour atteindre un palier puisque le vocabulaire est fini. La loi de Zipf[3] quantifie la répartition des fréquences des mots du vocabulaire (produit avec le rang approximativement constant).

A partir de cette information brute, de nombreux auteurs[7] dérivent de nouveaux vecteurs, cette fois-ci à valeurs réelles. La nouvelle matrice s'obtient par une transformation souvent linéaire du tableau de contingence initial. Les nouvelles valeurs servent éventuellement à la sélection du vocabulaire en employant un seuillage sur des statistiques en colonne de ces valeurs. Il a d'ailleurs été proposé des dizaines voir centaines de pondération possibles ; on emploie le terme pondération car usuellement, la transformation consiste souvent en une multiplication par un facteur dépendant seulement du mot v_j considéré, et identique pour l'ensemble des documents (ligne de D ou N). La plus usitée est certainement le "IDF"[7], correspondant au produit par "moins l'inverse du logarithme du nombre de documents comportant un mot donné" avec la fréquence brute. En indexation, un utilisateur pose une question $x_q = [N_{q1} N_{q2} \cdots N_{qJ}]^T$ et le système recherche les documents les plus proches en les classant par une fonction. Il s'agit ici de méthodes purement géométriques. Par exemple on calcule le produit scalaire $< x_q ; x_i >$, ou le cosinus qui a l'intérêt d'être borné $cos(x_q, x_i)$. Un document sera d'autant plus proche de la question qu'il partagera des mots avec elle. Il est clair que bien que très efficace et utilisée couramment dans la plupart des moteurs de recherche, cette approche ne prend pas en compte la complexité du langage en négligeant synonymie, polysémie et autres ; elle reste donc approximative. En outre, certains formalismes probabilistes aboutissent à des critères de comparaison semblables. Il paraît intéressant de chercher à lisser les vecteurs au moment de la comparaison, c'est ce que font les méthodes d'espace latent. Elles recherchent un sous-espace réduit qui permet de prendre en compte une information contextuelle sous-jacente. Il s'agit de projection linéaire de la matrice textuelle ou de réduction probabiliste à l'aide d'une variable aléatoire de classe à dimension $K << I$. Ces méthodes sont difficilement utilisables à grande échelle. La classification permet aussi d'accélérer l'accès aux documents en ne comparant x_q qu'à un nombre réduit de centres de classes. Au niveau des modèles probabilistes, certains réseaux bayésiens sont employés en classification[8], [9] et indexation [10], [11] [12]. Certaines de ces approches offrent des fonctionnalités d'interaction avec l'utilisateur. En outre, elles se résument parfois à des sommes pondérées de fonctions discriminantes linéaires.

[6] Une matrice creuse est une matrice numérique qui compte un grand pourcentage de cellules nulles.

1.4 Plan

Dans la suite de ce document, il n'y a aucune ambiguïté sur la notation x_i : x_i a toujours une forme vectorielle. Ce document de thèse s'articule en trois grandes parties :

1. Un chapitre sur l'état de l'art des méthodes classiques de réduction par analyses factorielles, classifications automatiques, modèles de mélange et une synthèse unifiée sur leur *généralisation* nommée surfaces principales discrètes,

2. Un chapitre sur les diverses méthodes originales développées dans le cadre de la thèse pour une analyse exploratoire des grands tableaux de contingence textuels,

3. Un chapitre d'applications et de simulations qui illustrent numériquement les méthodes sur des données simulées (corpus synthétique) ou réelles (corpus textuels), ainsi que visuellement par les grilles obtenues et les sorties d'un outil prototype de navigation interactive.

La conclusion résume les différentes contributions de cette recherche et donne des exemples de perspectives futures.

Chapitre 2

Etat de l'art en réduction par modèle de mélange de lois

Dans ce chapitre, nous nous intéressons à des tableaux de données numériques continues ou discrètes à I lignes et J colonnes. Dès que les dimensions de l'espace des lignes ou des colonnes dépassent trois, la visualisation du nuage des données est difficile sinon impossible. Nous présentons dans ce chapitre des méthodes de réduction de la dimension. Nous nous restreignons à des méthodes de classification et de projection utilisant des mélanges de lois de probabilité.

Nous allons présenter successivement des méthodes de classification par Nuées Dynamiques, des méthodes de projection linéaire par diagonalisation et des extensions non-linéaires des méthodes précédentes.

2.1 Méthodes de classification de type Nuées Dynamiques

2.1.1 Définition de la classification par partitionnement

Les méthodes de **classification automatique** non hiérarchique[1] construisent une partition[14] des données en K ensembles disjoints notés C_k, a priori non vides, si bien que $\mathcal{D} = \cup_{k=1}^{k=K} C_k$. Elles classent les données en K groupes avec les propriétés suivantes : chaque groupe contient au moins un objet et chaque objet appartient exactement à un groupe. Les questions à propos de la qualité de la classification, et du choix du nombre (inconnu) de classes sont difficiles dans le cas général. Il est possible de déterminer un nombre de classes à l'aide de critères statistiques. Ainsi, le partitionnement de I données $x_i \in \mathcal{D}$ aboutit à K ensembles les plus homogènes possibles pour le critère de classification considéré. Les ensembles volumineux de données imposent l'utilisation d'un algorithme de classification qui conduit à une

[1] Les méthodes de classification hiérarchique[13] partitionnent un ensemble de données sous la forme d'une suite de classes emboîtées dont la structure se représente comme un arbre binaire appelé dendrogramme.

solution sous-optimale mais acceptable, sachant que la meilleure solution est souvent inaccessible puisque le nombre de partitions à envisager pour un nombre[15] croissant de classes et de données devient rapidement gigantesque.

Plus formellement, on note c_{ik} les variables d'affectation aux classes qui affectent la donnée x_i à la k-ième classe. Une **partition** peut s'écrire :

$$\forall i,k\ c_{ik} \in \{0,1\}\ ,\ \sum_k c_{ik} = 1\ ,\ \text{et}\ \forall x_i \in C_k\ c_{ik} = 1 \tag{2.1}$$

La première condition place une donnée dans une classe unique tandis que la seconde rappelle que la donnée ne peut appartenir qu'à une seule classe à la fois. On en déduit que $k < I$. Si au lieu de coefficients binaires dans $\{0,1\}$, on emploie des valeurs dans $[0,1]$, on est dans le cas de la classification floue, et l'on note les c_{ik} flous par μ_{ik}. Dans un contexte probabiliste, on peut considérer ces μ_{ik} comme des probabilités a posteriori. L'intérêt d'un formalisme probabiliste parfois employé dans la suite est de pouvoir donner un sens au partition (au delà de la règle géométrique). Une **classification floue** s'écrit :

$$\forall i,k\ \mu_{ik} \in [0,1]\ ,\ \sum_k \mu_{ik} = 1 \tag{2.2}$$

Pour passer d'une classification floue à une partition, on affecte une donnée à la classe d'indice μ_{ik} maximum.

Le paragraphe suivant est consacré à l'algorithme des K-means, cas particulier de celui des Nuées Dynamiques[2].

2.1.2 Classification par minimisation de la variance-intra

L'algorithme des K-means[16, 17] est un cas particulier de celui des Nuées Dynamiques, pour des données dans $\mathcal{R}^J$. L'algorithme minimise le critère de variance-intra (solution sous-optimale). Si on appelle p_i une pondération de x_i, alors le critère de variance-intra s'écrit, pour des centres de classes notés m_k :

$$\sum_i p_i \sum_k c_{ik} ||x_i - m_k||^2_{\mathcal{R}_J} = \sum_k \sum_{x_i \in C_k} p_i ||x_i - m_k||^2_{\mathcal{R}_J} \tag{2.3}$$

Dans le cas classique où les pondérations sont toutes égales, $p_i = 1/I$, le critère devient :

$$\frac{1}{I} \sum_i \sum_k c_{ik} ||x_i - m_k||^2_{\mathcal{R}_J} = \frac{1}{I} \sum_k \sum_{x_i \in C_k} ||x_i - m_k||^2_{\mathcal{R}_J} \tag{2.4}$$

La minimisation de ce critère employé dans la suite, aboutit à un partitionnement de l'espace appelé diagramme de Voronoï. En effet, les classes solutions $\mathcal{C}_k$ sont telles que

[2]Les Nuées Dynamiques procèdent à la manière du K-means mais utilise un fonction quelconque pour rattacher une donnée à une classe qui n'est elle-même pas forcément représentée par la moyenne.

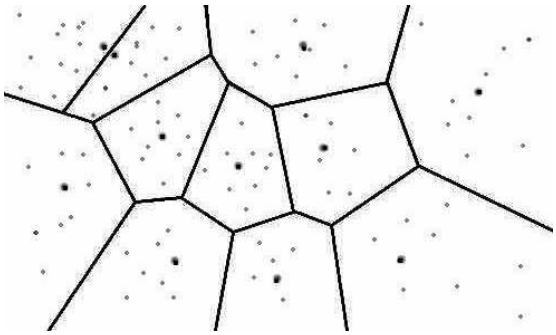


FIG. 2.1 – Cellules de Voronoï. Sur cet exemple, les centres sont les points en trait appuyé, et les données, les points plus petits. On distingue les frontières du pavage de Voronoï qui séparent les classes entre-elles. Dans chaque cellule, l'ensemble des individus est plus proche du centre de la cellule que de n'importe quel autre centre des autres cellules.

les individus qui leur sont affectés sont les plus proches du centre de classe m_k correspondant ; des auteurs ont montré que les frontières entre classes sont des portions d'hyperplans. Nous présentons deux algorithmes pour effectuer l'optimisation du critère. Le premier réévalue itérativement l'ensemble des centres à l'aide de l'échantillon $\mathcal{D}$ entier tandis que le second réévalue itérativement un centre à la fois après tirage d'un unique individu dans $\mathcal{D}$. On indice les variables par n pour expliciter le pas de l'algorithme. Pour les K-means, θ^n représente au pas n, les K centres de classes m_k.

2.1.2.1 Les K-means

Algorithme 1 *L'algorithme des K-means alterne une phase d'affectation et une phase de calcul des centres* $\boldsymbol{m} = \{m_k\}_k$. *Il minimise en introduisant la matrice de classification* $\boldsymbol{C} = (c_{ik})$, *le critère* $E(\boldsymbol{m}, \mathbf{C}|\mathcal{D}) = \sum_i \sum_k c_{ik} ||x_i - m_k||^2_{\mathcal{R}_J}$

Algorithme des K-MEANS

0. *Initialisation de* θ^0,
1. $Min_{\mathbf{C}} E(\boldsymbol{m}^{n-1}, \boldsymbol{C}|\mathcal{D}) = \sum_i \sum_k c_{ik} ||x_i - m_k^{n-1}||^2_{\mathcal{R}_J}$
   $\Rightarrow c_{ik}^n = \begin{cases} 1 & si\ ||x_i - m_k^{n-1}||_{\mathcal{R}_J}\ minimum, \\ 0 & sinon \end{cases}$
2. $Min_{\boldsymbol{m}} E(\boldsymbol{m}, \mathbf{C}^n|\mathcal{D}) = \sum_i \sum_k c_{ik}^n ||x_i - m_k||^2_{\mathcal{R}_J}$
   $\Rightarrow m_k^n = \frac{1}{I} \sum_k c_{ik}^n x_i$
3. $||\theta^n - \theta^{n-1}|| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
   sinon retour à 1.

L'algorithme consiste en l'itération de deux étapes, la première réaffecte chaque individu à sa classe la plus proche, pour des centres supposés fixes, de telle manière que $E(\mathbf{m^{n-1}}, \mathbf{C^n}|\mathcal{D}) \leqslant E(\mathbf{m^{n-1}}, \mathbf{C^{n-1}}|\mathcal{D})$, alors que la seconde recalcule les centres, en supposant les affectations fixes, si bien que $E(\mathbf{m^n}, \mathbf{C^n}|\mathcal{D}) \leqslant E(\mathbf{m^{n-1}}, \mathbf{C^{n-1}}|\mathcal{D})$, étant donnée la hessienne positive. Ainsi, chaque pas fait décroître ce critère borné par en dessous puisque positif : l'algorithme construit une suite de solutions convergentes vers un minimum (local et atteint) du critère. En effet, le critère peut ne prendre qu'un nombre fini de valeurs positives, comme le nombre de partitions de $\mathcal{D}$: le minimum (local) est atteint en un position stable.

La minimisation précédente revient à maximiser la variance interclasse car par la formule de Huygens, l'inertie totale s'écrit[3], avec m_G le centre de gravité de $\mathcal{D}$:

$$\underbrace{\sum_i p_i ||x_i - m_G||^2}_{\text{inertie totale}} = \underbrace{\sum_k \sum_i c_{ik} p_i ||x_i - m_k||^2}_{\text{inertie intra classes}} + \underbrace{\sum_k (\sum_i c_{ik} p_i) ||m_k - m_G||^2}_{\text{inertie inter classes}} \quad (2.5)$$

Les propriétés des estimations obtenues et du critère quadratique minimisé sont abondamment étudiées dans la littérature, en statistique ou en théorie de l'apprentissage, notamment dans la théorie de Vapnick. Le critère du K-means a donné lieu également à de nombreuses variantes dans le but d'améliorer la solution obtenue. Une classification posant explicitement la nature de la variabilité dans les classes emploie une distribution de classe arbitraire a priori, il s'agit des méthodes étudiées plus loin. Lorsque les données arrivent en flux continu, l'algorithme des K-means devient séquentiel pour recalculer les moyennes de classe après l'arrivée d'une donnée après l'autre.

2.1.2.2 Variante K-means séquentielle

[3]pour les moyennes $m_k = \sum_i c_{ik} p_i x_i$ et le centre de gravité du nuage $m_G = \sum_i p_i x_i$

Algorithme 2 *L'algorithme séquentiel des K-means ou K-means stochastique, optimise le critère quadratique à partir de sa dérivée locale ou gradient stochastique. Il minimise le même critère que précédemment pour un ensemble de données fini :*

Version K-MEANS séquentielle

0. *Initialisation de* θ^0,
1. *Sélection de* $x_{i_n} \in \mathcal{D}$,
2. $m_k^{n+1} = \begin{cases} m_k^n + \alpha_n(x_{i_n} - m_k^n) & \text{si } x_{i_n} \in \mathcal{C}_k^n \\ m_k^n & \text{sinon} \end{cases}$
3. $||\theta^n - \theta^{n-1}|| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
sinon retour à 1.

Pour avoir un bon comportement, il faut notamment que α_n *vérifie* $\sum_n \alpha_n = \infty$ *et* $\sum_n \alpha_n^2 \leq \infty$, *les conditions de Robbins-Monro[18] en approximation stochastique. Alors, à la limite, la vraie solution est atteinte.*

Fixer le α à une valeur faible permet d'obtenir des paramètres qui sont modifiés légèrement avec le temps. En analyse de données, pour un échantillon fini, la décroissance de α permet d'atteindre une solution par tirage séquentiel d'individus dans la population. L'algorithme des K-means même en version stochastique a le désavantage d'être très dépendant de l'initialisation, et de tomber facilement dans des minima locaux. C'est pourquoi, des auteurs ont proposé des moyens d'y remédier. Telles les techniques de recuit, des méthodes permettent d'atteindre l'optimum global et de ne pas s'arrêter à des solutions locales sous-optimales. Leur inconvénient majeur est de nécessiter un ralentissement logarithmique prohibitif de la vitesse de convergence des itérations. L'intérêt pratique est d'aboutir à de meilleurs partitionnements au prix d'une modification relativement aisée des procédures d'optimisation. Elles apportent également une réponse à la question de détermination du nombre de classes.

2.1.2.3 Recuit déterministe et simulé

Le recuit déterministe[19] consiste en l'ajout pour un critère donné d'un terme pénalisant sur la somme d'entropie des probabilités intervenant dans l'algorithme d'optimisation ; un terme de température T permet de pondérer l'importance de chacun des deux critères. Le recuit consiste alors à diminuer la température au cours des pas de l'algorithme jusqu'à l'annuler, et aboutir à une solution convenable. Par exemple, en classification par la variance-intra, on peut écrire :

$$E(\mathbf{m}|\mathcal{D}) = \sum_i \sum_k P(x_i, m_k)||x_i - m_k||^2 = \sum_i P(x_i) \sum_k P(m_k|x_i)||x_i - m_k||^2 \quad (2.6)$$

Les probabilités $P(m_k|x_i)$ s'identifient aux μ_{ik} précédents, et correspondent à des probabilités a posteriori de classe. Les probabilités $P(x_i)$ s'identifient aux

pondérations p_i précédentes et correspondent à la probabilité discrète d'une donnée. Alors, une procédure de recuit ajoute au critère E le terme d'entropie $H(\mathbf{m}|\mathcal{D}) = -\sum_i \sum_k P(x_i, m_k) \log P(x_i, m_k)$ pondéré par T, si bien que le critère minimisé devient $[E - T \times H](\mathbf{m}|\mathcal{D})$ qui s'optimise en itérant deux phases. La première évalue les distributions $P(m_k|x_i)$ à partir des centres courants. La seconde réestime ces centres. L'étude variationnelle de la fonction de coût intermédiaire fait apparaître des transitions de phase à des températures qui sont fonction des valeurs propres des matrices de covariance des données, conditionnellement à la variable de classe (assimilable à la variable prenant les K valeurs m_k). On note $\lambda(C_{\mathbf{x}})$ la plus grande valeur propre de la matrice de variance-covariance des données ; en effet, les changements de phase s'effectuent pour les plus grandes valeurs propres des matrices de variance-covariances de classe. L'algorithme de recuit déterministe s'écrit en notant α, un facteur de diminution géométrique de la température.

Pour éviter de calculer les valeurs propres (test sur T), Rose[19] préconise de conserver des centres dupliqués par légère perturbation, et d'observer leur séparation au cours de l'estimation et lors de la diminution de la température afin de détecter des changements de phase et l'apparition des nouvelles classes. L'intérêt de la méthode est évidemment de trouver les centres mais également leur nombre, de façon complètement automatique. Il s'agit d'une version déterministe du recuit simulé.

Algorithme 3

K-means par recuit déterministe

Limites	*température minimum* T_{min},
	nombre de classe maximum K_{max},
Init	$T > 2\lambda(C_{\mathbf{x}})$, $K = 1$, $m_1 = \sum_i P(x_i)x_i$, $P(m_1) = 1$,
Estimation	$\forall k, m_k = \frac{1}{P(m_k)} \sum_i P(x_i)P(m_k|x_i)x_i$,
	où $P(m_k|x_i) = \frac{P(m_k)e^{-\|\|m_k - x_i\|\|^2_{\mathcal{R}^J}/T}}{\sum_l P(m_l)e^{-\|\|m_k - x_i\|\|^2_{\mathcal{R}^J}/T}}$,
	et $P(m_k) = \sum_i P(x_i)P(m_k|x_i)$,
Test CV	*si pas arrêt retour à Estimation.*
Test T	*si* $T < T_{min}$, *dernière itération à T=0 et arrêt,*
Refroidissement	$T \leftarrow \alpha T, (0 < \alpha < 1)$,
	si $K < K_{max}$, *test* $(\forall k)$, *de la transition de phase,*
	et si vrai en k, dupliquer m_k *:* $m_{k+1} = m_k + \delta$,
	et $P(m_{k+1}) = P(m_k)/2$, $P(m_k) \leftarrow P(m_k)/2$, $K \leftarrow K + 1$,
Boucle	*Retourner à Estimation.*

Le recuit simulé (relaxation stochastique) est une méthode stochastique d'optimisation qui permet contrairement à la version déterministe de trouver le vrai optimum au prix d'une décroissance très lente de la température. Il a été introduit par Kirkpatrick[20] par analogie avec la physique statistique. La version stochastique réestime les paramètres aléatoirement de manière à obtenir une suite de solutions consécutives markoviennes. Il s'agit d'une chaîne de Markov convergeant vers une distribution stationnaire : la distribution se concentrant sur les états solutions. La méthode évite notamment d'avoir à calculer explicitement le paramètre de normalisation qui apparaît lorsque l'on s'intéresse à la probabilité de type de $exp(-E_\theta(\mathbf{X})/T)/\sum_i exp(-E_\theta(x_i)/T)$ pour un critère d'erreur E donné à minimiser pour θ. Il existe deux procédures principales. L'algorithme de Métropolis tire itérativement un nouvel état au hasard et modifie le paramètre s'il entraîne une diminution du critère (ΔE). Il est accepté également pour un tirage aléatoire réussi de probabilité $exp(-\Delta E/T)$. L'algorithme de Gibbs par contre calcule l'état le plus probable des paramètres à partir des estimations courantes, ce qui permet d'éviter des tirages inutiles. Généralement, une seule composante du vecteur des paramètres est modifiée ; elle est choisie ou bien par un parcours séquentiel sur l'ensemble des composantes, ou bien par tirage aléatoire. Enfin, une baisse inversement logarithmique de la température assure la convergence vers un optimum global, au prix d'un nombre d'itérations irréalisable en pratique. On préfère une décroissance exponentielle, qui permet notamment d'approcher la meilleure solution sans tomber trop rapidement dans une solution trop locale.

Le recuit procède à une optimisation probabiliste du critère qui reste de type partitionnement dur. L'algorithme des K-means emploie une règle géométrique d'affectation aux classes. Il paraît opportun de chercher à modéliser une distance entre données et centres pour qu'elle s'adapte à des nuages de densité non homogène. Les modèles de mélange de lois de probabilité, comme nous allons le voir, présentent de telles propriétés. Nous allons introduire le calcul du maximum de vraisemblance par *Expectation-Maximization* pour aboutir aux modèles de mélange et leur estimation notamment pour la réduction des données textuelles.

2.1.3 Maximum de vraisemblance et résolution EM

2.1.3.1 Maximum de vraisemblance

Cette partie introduit et décrit le formalisme des modèles et outils numériques utiles dans la suite. On se place dans un cadre probabiliste. On suppose que le corpus $\mathcal{D} = \{x_i\}_{i=1}^{i=I}$ représenté matriciellement par D, est un échantillon aléatoire généré par des variables aléatoires J-dimensionnelles $\mathbf{X} = \{X_i\}_{i=1}^{i=I}$ que nous considérerons le plus souvent identiquement distribuées et indépendantes les unes des autres (i.i.d.). La loi sous-jacente est inconnue ; elle sera supposée comme appartenant à une famille paramétrée par un vecteur $\theta \in \Theta$, dont il faudra rechercher le vraie valeur θ^0, estimée

par exemple par le maximum de vraisemblance, et notée $\hat{\theta}_I$ ou $\hat{\theta}$ en abrégé :

$$\hat{\theta}_I = argmax_\theta \, l(\theta|\mathcal{D}) = argmax_\theta \prod_i P_X(x_i|\theta) \tag{2.7}$$

On note Ω, l'espace des données, et Θ celui des paramètres. Le modèle correspondant est $\{\Omega, P_X(\bullet|\theta)\}_{\theta\in\Theta}$ où l'on suppose les données distribuées suivant la loi[4] précédente, pour un paramètre $\theta^0 \in \Theta$ inconnu et que l'on cherche à estimer. On s'intéressera essentiellement à une estimation par maximum de vraisemblance, c'est à dire l'estimateur $\hat{\theta}_I$ qui pour un échantillon i.i.d. de taille I, maximise la probabilité de l'échantillon ; souvent, la logvraisemblance, ou logarithme de la vraisemblance, est plus facile à étudier en raison des I produits transformés en une somme. Le maximum vérifie les équations de logvraisemblance puisque la fonction log est strictement croissante :

$$\nabla\mathcal{L}(\theta|\mathcal{D})|_{\theta=\hat{\theta}_I} = \sum_i [\nabla \log P_X(x_i|\theta)]_{\theta=\hat{\theta}_I} = 0 \tag{2.8}$$

L'intérêt de l'estimateur par maximum de vraisemblance est de présenter (sous certaines hypothèses de régularité) des propriétés intéressantes. Notamment, $\hat{\theta}_I \to \theta^0$ lorsque I devient très grand. Et, la distribution de l'écart est asymptotiquement gaussienne de matrice variance-covariance connue. Lorsque la solution n'a pas d'expression analytique, on emploie des algorithmes itératifs tel que l'algorithme Expectation-Maximization (ou EM) par exemple.

2.1.3.2 Algorithme EM

L'EM[22, 23] est une méthode itérative de maximisation de la vraisemblance. Cette méthode numérique emploie un conditionnement sur des variables inconnues. Si celles-ci sont bien choisies, elles simplifient la vraisemblance jointe et peuvent permettre une optimisation analytique. L'EM s'appuie sur des valeurs manquantes qualifiées de cachées ou latentes. Elles sont notées pour le cas des étiquettes inconnues de classes : $\mathbf{Z} = \{Z_1, Z_2, \cdots, Z_I\}$: ces variables aléatoires permettent d'écrire la distribution d'une donnée x_i conditionnellement à sa classe d'affectation, formellement Z_i et ainsi poser des distributions de classe, comme dans le modèle de mélange[5] de lois décrit dans la section suivante. L'algorithme EM cherche à maximiser la vraisemblance en marginalisant sur $\mathbf{Z}$:

$$\mathcal{L}(\theta|\mathcal{D}) = \log P(\mathbf{X}|\theta) = \log \int P(\mathbf{X}, \mathbf{Z}|\theta) \; dP_{\mathbf{Z}} \tag{2.9}$$

[4] On considèrera souvent des distributions multinomiales ou multivariées bernoulliennes, de Poisson qui sont les plus utilisées que ce soit en indexation ou en classification textuelle. Les deux premiers sont couramment utilisés[21] en recherche d'information textuelle : le multivarié bernoullien et le multinomial. Le premier ne prend pas en compte la taille des documents contrairement au second. Le troisième est également employé et fait notamment l'objet de plus anciens travaux en indexation où il semble donner des résultats peu robustes.

[5] Ici, Z_i permet de formaliser la notion de classe en prenant ses valeurs dans $\mathcal{Z} = \{1, 2, \cdots, K\}$ codant des étiquettes de classes, si bien que "$Z_i = k$ *signifie que* X_i *est dans la k-ième classe*".

En reprenant Russel dans son article[24], on a :

$$\begin{aligned}
&\mathcal{L}(\theta|\mathcal{D}) - \mathcal{L}(\theta^n|\mathcal{D}) \\
&\qquad = \log P(\mathbf{X}|\theta) - logP(\mathbf{X}|\theta^n) = \log \tfrac{P(\mathbf{X}|\theta)}{P(\mathbf{X}|\theta^n)} \\
&\qquad = \log \int \tfrac{P(\mathbf{X}|\mathbf{Z},\theta)P(\mathbf{Z}|\theta)}{P(\mathbf{X}|\theta^n)} dP_{\mathbf{Z}} = \log \int \tfrac{P(\mathbf{X}|\mathbf{Z},\theta)P(\mathbf{Z}|\theta)}{P(\mathbf{X}|\theta^n)} \tfrac{P(\mathbf{Z}|\mathbf{X},\theta^n)}{P(\mathbf{Z}|\mathbf{X},\theta^n)} dP_{\mathbf{Z}} \\
&\qquad \geq \int P(\mathbf{Z}|\mathbf{X},\theta^n) \log \tfrac{P(\mathbf{X}|\mathbf{Z},\theta)P(\mathbf{Z}|\theta)}{P(\mathbf{X}|\theta^n)P(\mathbf{Z}|\mathbf{X},\theta^n)} dP_{\mathbf{Z}} \\
&\qquad = \underbrace{\int P(\mathbf{Z}|\mathbf{X},\theta^n) \log P(\mathbf{Z},\mathbf{X}|\theta)\, dP_{\mathbf{Z}}}_{\mathcal{Q}(\theta|\theta^n)} - \underbrace{\int P(\mathbf{Z}|\mathbf{X},\theta^n) \log P(\mathbf{Z},\mathbf{X}|\theta^n)\, dP_{\mathbf{Z}}}_{CTE(\theta^n)}
\end{aligned} \tag{2.10}$$

On a noté $CTE(\theta^n)$ le terme qui n'est pas fonction de notre inconnue θ, et considéré constant. L'inégalité précédente s'obtient par celle de Jensen. Soit finalement avec la bonne notation :

$$\mathcal{L}(\theta|\mathcal{D}) \geq \mathcal{L}(\theta^n|\mathcal{D}) + \mathcal{Q}(\theta|\theta^n) - CTE(\theta^n) \tag{2.11}$$

Donc la valeur θ^{n+1} qui maximise $\mathcal{Q}(\theta|\theta^n)$ en θ fait croître la vraisemblance. En effet, $\mathcal{Q}(\theta^{n+1}|\theta^n) \geq \mathcal{Q}(\theta^n|\theta^n) = CTE(\theta^n)$, qui assure $\mathcal{L}(\theta^{n+1}|\mathcal{D}) \geq \mathcal{L}(\theta^n|\mathcal{D})$. On se réfèrera à [22] pour une démonstration approfondie. On en déduit finalement l'algorithme Expectation-Maximization :

Algorithme 4 *L'EM effectue une maximisation de la logvraisemblance $\mathcal{L}(\theta|\mathcal{D})$ par :*

Algorithme EM

Init	*Initialisation de* θ^0,
Pas E	$\mathcal{Q}(\theta|\theta^n) = \mathbb{E}_{\mathbf{z}|\mathcal{D},\theta^n}[\log P(\mathbf{X},\mathbf{Z}|\theta)]$,
Pas M	$\theta^{n+1} = argmax_\theta\ \mathcal{Q}(\theta|\theta^n)$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
	sinon retour au pas E.

A chaque itération n, l'algorithme est assuré de faire croître (ou garder constant) la vraisemblance. A la convergence, l'algorithme atteint un optimum (local) de $\mathcal{L}(\theta|\mathcal{D})$ en $\hat{\theta}$.

Il se peut que cela ne soit pas un maximum dans certains cas. Il faut éventuellement vérifier en calculant la hessienne. L'algorithme EM a donné lieu à de nombreux travaux pour évaluer empiriquement ou théoriquement l'intérêt de l'utiliser au lieu d'une maximisation classique par gradient (Newton) notamment. Brièvement, l'algorithme a dans certains cas une vitesse de convergence meilleure qu'un gradient classique. En effet, il opère comme un algorithme de gradient, mais régularisé par une matrice particulière. Il ne nécessite pas l'estimation d'une hessienne qui peut s'avérer difficile à évaluer en pratique. En outre, l'EM prend en compte de façon élégante les questions

de contraintes sur les paramètres. Pour les détails, le lecteur se référera à [25, 26].

Certaines approximations s'avèrent nécessaires dans un cas analytique ou combinatoire insoluble. Ainsi, la phase *Maximization* peut s'effectuer par exemple par gradient. On parle de GEM pour EM Généralisé ; après le pas *Expectation*, le GEM effectue avec α_r décroissante en r, pour r_{max} pas au total ($\theta^{n+1} = \theta^{n+1,r_{max}}$), et éventuellement un pas en r unique :

$$\begin{array}{l}\text{Pas M :}\\ \theta^{n+1,r} = \theta^{n+1,r-1} + \alpha_r \times \nabla \left[\mathcal{Q}(\theta|\theta^n)\right]_{\theta=\theta^{n+1,r-1}}\end{array}$$

De même, la phase *Expectation* peut être incalculable en pratique. Des auteurs proposent alors d'employer un calcul par Monte-Carlo. C'est-à-dire qu'au lieu du calcul *Expectation* exact, on calcule une valeur approchée par sommation à partir d'un échantillon a posteriori calculé par simulation :

$$\begin{array}{l}\text{Pas E :}\\ \bar{\mathcal{Q}}(\theta|\theta^n) = \sum_l \sum_i \log P(z_{il}^n, x_i|\theta) \text{ où } \forall i, l,\ z_{il}^n \sim P(Z|x_i, \theta^n)\end{array}$$

Il existe également des versions de l'EM qui sont qualifiées de stochastiques (SEM et SAEM). Proposées par Celeux et Diebolt, elles affectent de façon aléatoire les données aux centres de classe. D'autres méthodes approchées sont possibles, notamment en supprimant certaines dépendances stochastiques entre variables pour rendre possible un calcul analytique. On parle de méthodes variationnelles. Des méthodes formulant des lois a priori sur les paramètres sont aussi envisageables. On parle de méthodes MCMC pour Markov Chain Monte Carlo ou méthodes de simulation bayésienne. En pratique, l'EM donne des résultats assez bons pour peu que l'initialisation ne soit pas trop éloignée du maximum global et que le modèle soit réaliste. On estime notamment les mélanges de lois par EM (voir suite), où le label (ou étiquette) de classe est justement inconnu.

2.1.4 Mélange de lois et vraisemblance classifiante floue

On peut écrire un mélange de loi, où chaque loi $P(X|Z = k; \theta)$ ne dépend en réalité (pour les mélanges étudiés) que de ses paramètres a_k. La variable Z est appelée la variable de classe et prend ses valeurs dans $\mathcal{K} = \{1, 2, \cdots, K\}$:

Définition 7 *Un mélange de loi pour la v.a. aléatoire observée X_i de réalisation x_i, avec l'étiquette de classe Z_i de réalisation $z_i \in \mathcal{K}$, a pour expression :*

$$P(X_i = x_i|\theta) = \sum_{k\in\mathcal{K}} P(Z_i = k|\theta) P(X_i = x_i|Z_i = k; \theta) \tag{2.12}$$

On n'écrit plus les v.a. dans la suite pour simplifier. L'ensemble des Z_i est une v.a. notée $\mathbf{Z}$, de réalisations $\mathbf{z}$. Les probabilités $\pi_k = P(Z_i = k|\theta)$ sont appelées coefficients mélangeants ou paramètres a priori du k-ième facteur du mélange. Elles s'estiment

sur un échantillon généralement comme la fraction d'individus appartenant à la classe correspondante (ou plus probable). Les probabilités $P(x_i|k;\theta) = P(x_i|a_k)$ sont les distributions pour chacun des facteurs. Finalement $\theta = (a_1, a_2, \cdots, a_K, \pi_1, \pi_2, \cdots, \pi_K)$. Voici l'expression de la logvraisemblance :

Définition 8 *En supposant l'ensemble des documents générés i.i.d. par une v.a. X, la logvraisemblance de cet échantillon s'écrit :*

$$\mathcal{L}(\theta|\mathcal{D}) = \log \prod_{i\in\mathcal{I}} P(x_i|\theta) = \sum_{i\in\mathcal{I}} \log \sum_{k\in\mathcal{K}} \pi_k P(x_i|a_k)$$

Elle s'obtient par marginalisation sur les Z_i, de leur loi jointe multinomiale, dans la vraisemblance :

Définition 9 *La logvraisemblance complétée s'écrit :*

$$\mathcal{L}(\theta, \mathbf{z}|\mathcal{D}) = \sum_{i\in\mathcal{I}} \log \left(P(z_i|\theta) P(x_i|z_i;\theta)\right)$$

Si les classes sont bien séparées dans $\mathcal{L}(\theta|\mathcal{D})$, alors à chaque document correspond une unique classe. Si on note $\mathbf{C} = (c_{ik})$ la matrice de classification ($c_{ik} \in \{0,1\}$ et $\forall i \sum_k c_{ik} = 1$), on peut écrire :

Définition 10 *La logvraisemblance classifiante s'écrit :*

$$\mathcal{L}(\theta, \mathbf{C}|\mathcal{D}) = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} c_{ik} \log \left(\pi_k \, P(x_i|a_k)\right) \tag{2.13}$$

Il s'agit d'une reformulation déterministe de $\mathcal{L}(\theta, \mathbf{z}|\mathcal{D})$ à l'aide de la matrice de classification. L'optimisation du critère correspond à un algorithme de classification ; on parle de CEM[27] pour *Classifying EM*; cet algorithme est équivalent à l'ajout à l'EM d'un pas d'affectation aux classes avant celui d'estimation, l'affectation classant une donnée dans la classe qui lui est la plus probable a posteriori. D'ailleurs[27],

Propriété 1
La maximisation du critère de vraisemblance classifiante pour des gaussiennes équiprobables à variances sphériques identiques est équivalente à la minimisation du critère des moyennes mobiles ou K-means.
□

On peut également définir l'EM d'un mélange comme l'optimisation d'un critère[28] à paramètres flous particuliers :

Définition 11 *La logvraisemblance floue s'écrit avec μ la matrice de classification floue :*

$$\mathcal{L}(\theta, \mu|\mathcal{D}) = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} \mu_{ik} \log \left(\pi_k \, P(x_i|a_k)\right) - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}} \mu_{ik} \log \mu_{ik} \tag{2.14}$$

Chaque coefficient flou résultant correspond à un degré d'appartenance à la classe et s'interprète également comme la probabilité a posteriori de classe, $\mu_{ik} = t_k^n(x_i)$ pour :

$$t_k^n(x_i) = \frac{\pi_k^n P(x_i|a_k^n)}{\sum_l \pi_l^n P(x_i, a_l^n)} = P(k|x_i;\theta^n) \tag{2.15}$$

puisque la solution s'identifie à celle de l'EM, et en posant en exposant le pas n.
La fonction Q dans le cas du mélange s'écrit :

$$\mathcal{Q}(\theta|\theta^n) = \sum_i \sum_{k\in\mathcal{K}} t_k^n(x_i) \log\left(\pi_k P(x_i|a_k)\right) \tag{2.16}$$

On en déduit l'algorithme de résolution de θ par EM d'un mélange :

Algorithme EM pour un mélange

Init	Initialisation de θ^0,
Pas E	$\forall k\,,\ \forall i\,,\ t_k^n(x_i) = \pi_k^n P(x_i|a_k^n)/\sum_l \pi_l^n P(x_i|a_l^n)$,
Pas M	$\forall k\,,\ \pi_k^{n+1} = \sum_i t_k^n(x_i)/I$,
	$\forall k\,,\ a_k^{n+1} = argmax_{a_k} \sum_i \sum_{k\in\mathcal{K}} t_k^n(x_i) \log P(x_i|a_k)$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ alors $\hat{\theta} = \theta^n$, sinon retour au pas E.

Dans le cas du CEM maximisant $\mathcal{L}(\theta, \mathbf{z}|\mathcal{D})$ (ou $\mathcal{L}(\theta, \mathbf{C}|\mathcal{D})$), on a :

Algorithme CEM pour un mélange

Init	Initialisation de θ^0,
Pas E	$\forall k\,,\ \forall i\,,\ t_k^n(x_i) = \pi_k^n P(x_i|a_k^n)/\sum_l \pi_l^n P(x_i|a_k^n)$,
Pas C	$\forall k\,,\ \forall i\,,\ c_{ik}^n = \begin{cases} 1 & \text{si } k = argmax_k\ t_k^n(x_i) \\ 0 & \text{sinon} \end{cases}$,
Pas M	$\forall k\,,\ \pi_k^{n+1} = \sum_i c_{ik}^n/I$,
	$\forall k\,,\ a_k^{n+1} = argmax_{a_k} \sum_i c_{ik}^n \log P(x_i|a_k)$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ alors $\hat{\theta} = \theta^n$, sinon retour au pas E.

Nous illustrons les algorithmes dans le cas gaussien pour des x_i supposés continus. Dans cet exemple, les distributions de classes sont posées gaussiennes (normales multidimensionnelles), si bien que :

Exemple 6
Loi de X_i considérée comme un mélange de gaussiennes
On pose $X_i \sim \sum_{k\in\mathcal{K}} \pi_k f_k(x_i|a_k)$ où $f_k(x_i|a_k)$ est une densité de loi normale de paramètres $a_k = (m_k, \Sigma_k)$ où m_k et Σ_k sont respectivement l'espérance et la matrice de variance-covariance dans $f_k(x_i|a_k) \propto |\Sigma_k|^{-\frac{1}{2}} exp(-\frac{1}{2}(x_i - m_k)^T \Sigma_k^{-1}(x_i - m_k))$. On en déduit l'algorithme suivant, après calculs algébriques sur $Q(\theta|\theta^n)$:

EM d'un mélange de gaussiennes

Init	*Initialisation de* θ^0,
Pas E	$\forall k, i\ t_k^n(x_i) \propto \pi_k^n f_k(x_i\|a_k^n)$
Pas M	$\forall k\ m_k^{n+1} = \sum_i \frac{t_k^n(x_i)}{\sum_i t_k^n(x_i)} x_i$ *et* $\pi_k^{n+1} = \frac{\sum_i t_k^n(x_i)}{I}$
	$\forall k\ \Sigma_k^{n+1} = \frac{\sum_i t_k^n(x_i)(x_i - m_k^n)(x_i - m_k^n)^T}{\sum_i t_k^n(x_i)}$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$, *sinon retour au pas E.*

Nous illustrons l'algorithme dans deux cas multinomiaux. Pour le premier cas, les lignes du tableau de contingence D, les données, sont partitionnées en K classes multinomiales. Dans le second cas, l'ensemble des cellules suit une loi multinomiale paramétrée par un modèle de mélange. Ces méthodes décomposent un tableau de contingence à l'aide des modèles de mélange, et donnent d'excellents résultats dans le domaine des données textuelles.

Exemple 7

Loi de X_i considérée comme un mélange de multinomiales

On pose $X_i \sim \sum_k \pi_k \prod_j P_{j|k}{}^{N_{ij}}$, *le mélange de lois multinomiales ordonnées[29] qui sont employées ici pour éviter de manipuler le coefficient mélangeant et le paramètre* $|x_i|$ *de taille des textes. On en déduit l'algorithme de résolution du maximum de vraisemblance par EM. Une classification des lignes s'obtient par une dernière itération de type CEM, en affectant chaque document à la classe de maximum a posteriori, proportionnel à* $\hat{\pi}_k \prod_j \hat{P}_{j|k}^{N_{ij}}$ *pour* $\hat{\theta} = (\hat{\pi_1}, \hat{\pi_2}, \cdots, \hat{\pi_K}, \hat{a}_1, \hat{a}_2, \cdots, \hat{a}_K)$ *avec* $\hat{a}_k = (\hat{P}_{1|k}, \hat{P}_{2|k}, \cdots, \hat{P}_{J|k})$. *Il s'agit en réalité d'un modèle sur les séquences des mots des textes avec un historique réduit à zéro.*

Exemple 8

Loi de la variable aléatoire ayant générée le tableau de contingence D, et considérée comme une loi multinomiale paramétrée par un mélange

Ce modèle récemment introduit par Thomas Hofmann[30] s'appelle PLSA (Probabilistic Latent Analysis). Au lieu de considérer des lignes multinomiales, c'est l'ensemble du tableau que l'on suppose multinomial.
Le modèle s'écrit : $D \sim \mathcal{M}(N_{\bullet\bullet}, \{P_{ij} = \sum_k P_k P_{j|k} P_{i|k}\}_{i,j})$. *D'où :*

$$L(\theta|\mathcal{D}) = \log\Big(\prod_{i=1}^{i=I}\prod_{i_j=1}^{i_j=|x_i|} P_{i,i_j}\Big) = \log\Big(\prod_{i=1}^{i=I}\prod_{j=1}^{j=J} P_{ij}{}^{N_{ij}}\Big) = \sum_{i=1}^{i=I}\sum_{j=1}^{j=J} N_{ij}\ \log P_{ij} \quad (2.17)$$

Pour estimer le modèle, l'auteur applique l'EM et ainsi calcule les distributions conditionnelles. L'algorithme s'écrit en notant θ *le vecteur des paramètres estimés :*

Algorithme du PLSA

Init	*Initialisation de* θ^0,
Pas E	$\forall k,i,j\ P^n_{k\|i,j} \propto P^n_k P^n_{i\|k} P^n_{j\|k}$,
Pas M	$\forall k\ P^{n+1}_k = \frac{1}{N_{\bullet\bullet}} \sum_{i,j} N_{ij} P^n_{k\|i,j}$,
	$\forall k,i,j\ P^{n+1}_{j\|k} = \frac{\sum_i N_{ij} P^n_{k\|i,j}}{\sum_l \sum_i N_{il} P^n_{k\|i,l}}$; $P^{n+1}_{i\|k} = \frac{\sum_j N_{ij} P^n_{k\|i,j}}{\sum_l \sum_j N_{ij} P^n_{k\|i,j}}$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$, *sinon retour au pas E.*

Ici, on peut écrire pour le vecteur de paramètres $\hat{\theta}$, $\hat{\pi}_k = \hat{P}_k$ et $\hat{a}_k = (\hat{P}_{i|k})_i \cup (\hat{P}_{j|k})_j$, mais il ne s'agit pas d'un mélange de définition classique. La méthode PLSA réalise une réduction non-linéaire des données par le biais de la variable cachée sur laquelle se distribuent à la fois le corpus et le vocabulaire. Elle permet d'obtenir des résultats plus performants pour les benchmarks de Hofmann[31], que la méthode linéaire (purement algébrique) alternative historiquement ultérieure et présentée dans la section suivante, et appelée SVD pour décomposition en valeurs singulières. La matrice est également décomposée en un produit de sous matrices par le PLSA.

Dans la partie perspective en conclusion, nous proposons une formulation du modèle PLSA en Chaîne de Markov Cachée, et que l'on appelle HMM-PLSA, afin de prendre en compte l'enchaînement des classes cachées dans le cas du langage écrit vu comme un enchaînement de concepts.

2.2 Méthodes de projection linéaire par SVD

Les méthodes de projection planaire construisent une représentation synthétique de la structure d'un corpus de données multidimensionnelles sous la forme d'un plan de projetés. Dans un espace euclidien, les méthodes de statistique exploratoire multivariées cherchent des directions expliquant au maximum les origines de l'inertie du nuage de données. Ce sont [3] :

> *"Les méthodes factorielles, largement fondées sur l'algèbre linéaire, produisent des représentations graphiques sur lesquelles les proximités géométriques usuelles entre point-lignes et entre points-colonnes traduisent les associations statistiques entre lignes et colonnes. C'est à cette famille de méthodes qu'appartient l'analyse des correspondances[...]".*

Ces méthodes algébriques sont basées sur la décomposition d'une matrice sous la forme d'un produit de matrices orthogonales et diagonales. Le théorème suivant est le principal outil des méthodes :

Théorème 1 *[32]*
Toute matrice D de rang K admet une décomposition en valeurs singulières :

$\forall D \in \mathcal{R}^I \times \mathcal{R}^J$,
$\exists U \in \mathcal{R}^I \times \mathcal{R}^K$, $V \in \mathcal{R}^J \times \mathcal{R}^K$ *à colonnes orthogonales et,*
$\exists \Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_K)$ *t.q.,*

$$D = U\Sigma V^T \tag{2.18}$$

Dans la présentation classique, les méthodes factorielles s'appuient sur la diagonalisation d'une matrice symétrique obtenue par produit d'une matrice rectangulaire par sa transposée DD^T ou D^TD, dont la décomposition s'obtient facilement à partir du théorème. Ainsi, l'Analyse en Composantes Principales[13] ou ACP travaille avec la distance euclidienne, alors que l'Analyse des Correspondances ou AFC emploie une métrique adaptée aux tableaux de contingence. Nous présentons l'AFC dans la suite ainsi qu'une méthode d'approximation de matrice par SVD tronquée. L'ensemble de ces méthodes construisent une décomposition de matrice puis s'intéressent aux plus grandes valeurs d'éléments diagonaux ou valeurs propres de la matrice Σ. En effet, au sens des moindres-carrés, reconstruire D en supprimant les petites valeurs propres, au rang K fournit la matrice la plus proche de D, et de rang K, pour la norme de Frobenius (somme des carrés de l'ensemble des cellules de la matrice différence).

2.2.1 Analyse Factorielle des Correspondances (AFC)

L'AFC[33, 13] a été développée notamment par Benzécri, Escofier au début des années 1960 en France. C'est une analyse factorielle dotée d'une métrique particulière et adaptée à des données qualitatives. La métrique employée est celle du χ^2. Elle présente le principe d'équivalence distributionnelle formulé par Benzécri. Elle reste inchangée si des modalités de profils identiques sont regroupées.

Définition 12 *Pour comparer des profils-lignes (distribution conditionnée par la marginale de la ligne), la* ***distance du*** χ^2 *s'écrit :*

$$d_{\chi^2}(P_{i_1\bullet}, P_{i_2\bullet})^2 = ||f_{i_1\bullet} - f_{i_2\bullet}||^2_{\chi^2} = \sum_{j=1}^{j=J} \frac{1}{f_j}(f_{j|i_1} - f_{j|i_2})^2 \tag{2.19}$$

Dans le cas général, une méthode factorielle[13, 34] maximise dans l'espace des variables (ou des individus) l'inertie du nuage projeté sur une direction ou vecteur u :

Définition 13 ***L'analyse factorielle avec métrique M et pondération P***
Le nuage est transformé par une métrique M donnée. A chaque point est appliqué un poids, élément de la matrice diagonale de poids P. En outre, u est supposé normé pour M, i.e., $u^TMu = 1$. *L'inertie projetée du nuage, pour M et P fixées, s'écrit alors* u^TMA^TPAMu. *La maximisation sous la contrainte pour u aboutit après introduction du lagrangien* λ *à :*

$$\exists \lambda \in \mathcal{R}_+ \text{ t.q. } A^TPAMu = \lambda u \tag{2.20}$$

Le théorème énoncé précédemment (dans sa version pour matrice symétrique) permet de déduire l'existence de λ, et de le caractériser comme une valeur propre de la matrice $A^T PAM$. En prémultipliant par $u^T M$, il est clair qu'il s'agit également de l'inertie projetée, donc de la plus grande valeur propre. Finalement, les méthodes factorielles construisent une suite u_k de vecteurs orthogonaux (à ceux les précédant) maximisant l'inertie (restante). Suivant la métrique, la nature des données, et la matrice poids, la projection présente certaines propriétés statistiques intéressantes.

L'AFC travaille sur la matrice des profils ligne ou colonne. Ainsi, pour la matrice des données $D \in {\mathbb{N}_+}^{I \times J}$, matrice des comptage d'occurrence de mots, on écrit :

	$\mathcal{R}^J$	$\mathcal{R}^I$
Données	$A = {D_I}^{-1} D$	$A = {D_J}^{-1} D^T$
Métrique	$M = {D_J}^{-1}$	$M = {D_I}^{-1}$
Poids	$P = D_I$	$P = D_J$
Diagonaliser	$D^T {D_I}^{-1} D {D_J}^{-1}$	$D {D_J}^{-1} D^T {D_I}^{-1}$

L'AFC décompose le χ^2 d'indépendance du tableau de contingence correspondant. L'inertie du tableau diagonalisé, soit la somme des valeurs propres, vérifie sous hypothèse d'indépendance :

$$N_{\bullet\bullet}(trace[D^T {D_I}^{-1} D {D_J}^{-1}] - 1) \sim \chi^2_{(I-1)(J-1)} \tag{2.21}$$

Comme la matrice des profils lignes somme en ligne à l'unité, la solution est dans un sous-espace de taille I-1 ou J-1 suivant l'espace d'origine. Le barycentre est en effet le vecteur propre pour la valeur propre 1, alors que l'ensemble des autres valeurs propres sont de module inférieur à l'unité. Des indicateurs pour guider à l'interprétation existent : contribution[6], qualité[7], et signe de la projection. L'AFC est aujourd'hui très utilisée en pratique par les statisticiens, notamment en analyse des données textuelles. Les analystes l'emploient essentiellement dans une optique d'étude exploratoire des données, par visualisation directe sur les plans factoriels :

Définition 14
Un plan factoriel est la projection des données sur l'hyperplan formé de la moyenne empirique des données $\sum_i p_i x_i$ pour les pondérations p_i et de deux directions factorielles quelconques $\{u_{k_1}, u_{k_2}\}$

Il est important de noter la propriété de double représentation barycentrique de l'AFC qui autorise la superposition des projections des deux espaces (variables et individus) et l'interprétation des proximités. Il est également possible d'effectuer par exemple une réduction en conservant seulement les premiers facteurs, juste avant un algorithme de classification comme les K-means. Les classes obtenues peuvent être affichées sur

[6] contribution : pourcentage d'inertie d'une donnée
[7] carré du cosinus entre un vecteur de donnée et la direction de projection considérée

les plans factoriels. La principale limite de la méthode est d'ordre algorithmique : la diagonalisation d'une grande matrice (peu dense de surcroît) demeure un problème difficile.

2.2.2 Latent Semantic Analysis (LSA)

Le LSA[35], méthode géométrique proposée par Berry, consiste en la reconstruction de la matrice des données par SVD tronquée. Elle s'effectue sur de grands tableaux creux de données (textuelles en général), de l'ordre de plusieurs centaines de milliers de lignes ou colonnes. Le plus performant algorithme de LSA est en $O(I^2\tilde{K}^3)$, où on note $\tilde{K}$ le rang de la matrice tronquée. Le LSA n'est pas sans rappeler le PLSA auquel il a donné son nom. Il est difficile de savoir s'il a été conservé un nombre suffisant de valeurs propres. Toutefois, cette méthode améliore[7] sensiblement les calculs de similarités entre vecteurs de données. Cela est généralement vrai si les dimensions sont raisonnables, i.e. si la SVD est calculable. Quelques papiers ont été proposés pour expliquer (sous des hypothèses peu générales) le succès de la méthode. Des auteurs construisent également des projections planaires à partir des directions orthogonales du LSA, mais elles n'ont pas d'interprétation statistique comme en ACP. Pourtant les directions orthogonales de U et V s'identifient à celle d'une méthode d'analyse factorielle, l'Analyse en Composantes Principales (ACP), décrite dans la partie suivante. On présente également une classe de généralisations non-linéaires des plans factoriels : les surfaces principales et leur discrétisation approchée.

2.3 Méthode ACP et extensions non-linéaires

Cette partie explore différentes méthodes relativement récentes de projection d'une distribution par le biais de variables cachées contraintes sur un plan dans $\mathcal{R}^2$. Le but est de présenter d'un point de vue algorithmique la problématique de projection d'une distribution sur un plan. On peut également voir ces méthodes comme des estimations de densités paramétriques contraintes par une variable conditionnelle. Cette variable représente le plan qui est soit discrétisé et modélisé à l'aide de lois de Dirac, soit continu et modélisé à l'aide d'une distribution gaussienne dans $\mathcal{R}^2$.

2.3.1 ACP et surfaces principales

Dans cette section, x_i est un vecteur multivarié de $\mathcal{R}^J$, réalisation de la variable aléatoire X. On rappelle qu'une surface est généralement une fonction f de $\mathcal{R}^2$ dans R^J. A partir de cette définition, Hastie a proposé une nouvelle méthode en construisant des **Surfaces Principales**, ayant des propriétés particulières dans un espace probabilisé :

Définition 15
Une surface principale[36, 37] f^{SP} *est une courbe continue paramétrée* $\mathcal{C}^\infty$ *:*

$$f^{SP}(\xi) = \mathbb{E}_{P_X}\left[X|g^{SP}(X) = \xi\right] \quad \forall \xi \in \Xi \subseteq \mathcal{R}^2 \tag{2.22}$$

On note g une fonction de projection qui vérifie :

$$g^{SP}(X) = sup_{\xi_1 \in \Xi} \left\{ \xi : ||X - f(\xi)||_{\mathcal{R}^J} = inf_{\xi'} ||X - f^{SP}(\xi')||_{\mathcal{R}^J} \right\} \tag{2.23}$$

Ces surfaces sont des généralisations non-linéaires des plans factoriels de l'ACP, donc elles tiennent compte des corrélations non-linéaires. En effet, elles étendent simplement au cas 2-dimensionnel les **Courbes Principales** qui sont elles-mêmes des généralisations des directions principales de l'ACP. Une courbe principale[8] est un point critique de D(P,f), l'espérance par la distribution P_X de la distance euclidienne entre un point et sa projection sur la courbe f :

Théorème 2 *[36]*

$$\left. \frac{\partial \{ \mathbb{E}_{P_X} (||X - f_h^{CP}(g^{CP}(X))||^2_{\mathcal{R}^J}) \}}{\partial h} \right|_{h=0} = 0 \quad avec\ f_h^{CP} = f^{CP} + hg\,, \quad \forall g \in \mathcal{G}_B \tag{2.24}$$

Cette propriété de point critique appelée d'auto-consistance rappelle celle de moindres-carrés de l'ACP linéaire. Les courbes principales sont par exemple les facteurs de l'ACP dans le cas de distributions gaussiennes (ou sphériques). Un plan factoriel est une surface principale linéaire puisqu'il minimise la distance des points à leurs projetés. L'analyse en composantes principales cherche à projeter les données sur un sous-espace de dimension K réduite. On note celui-ci H_U, défini par un point X^0, et une base de K vecteurs $\{u_1, u_2, \cdots, u_K\}$. Une donnée projetée s'écrit $f^{ACP}(g^{ACP}(x_i)) = \bar{X}_I + U_K g^{ACP}(x_i)$, où $U_K = [u_1|u_2|\cdots|u_K]$ est une matrice de taille J*K à colonnes orthogonales. Ces dernières correspondent, comme on va le montrer, aux premiers vecteurs propres (on prend les plus grandes valeurs propres) qui diagonalisent la matrice de variance-covariance des données. $\bar{X}_I$ est le centre de gravité du nuage des données. La fonction de projection s'écrit : $g^{ACP}(x_i) = U_K^T(x_i - \bar{X}_I)$ comme les coordonnées de x_i dans la nouvelle base. Le critère de moindres-carrés s'écrit[9] pour un échantillon de données, avec $f_h^{ACP} = f^{ACP} + hg\,, \quad \forall g \in \mathcal{G}_L$:

$$\left. \frac{\partial \{ \frac{1}{I} \sum_i ||x_i - f_h^{ACP}(g^{ACP}(x_i))||^2_{\mathcal{R}^J} \}}{\partial h} \right|_{h=0} = 0 \tag{2.25}$$

Preuve

Nous traitons le cas général d'une distribution quelconque en employant l'espérance au lieu de la moyenne empirique (de l'échantillon $\mathcal{D}$) et calculer la solution optimale directement.

$$Min_{X^0, [u1|u_2|\cdots|u_K]} \mathbb{E}||X - f^{ACP}(g^{ACP}(X))||^2_{R^J} \tag{2.26}$$

[8] Si $\mathcal{G}_B$ est la classe des courbes $\mathcal{C}^\infty$ paramétrées sur $\mathcal{S} \subseteq \mathcal{R}^1$ avec $||g|| \leq 1$ et $||\dot{g}|| \leq 1$, alors, f^{CP} est une courbe principal de P, ssi f^{CP} est un point critique de la fonction de distance D(P,f) pour les perturbations dans $\mathcal{G}_B$.

[9] On note $\mathcal{G}_L$ l'ensemble des fonctions linéaires sur le sous-espace span(D)

Il est clair que pour la v.a. X,

$$f^{ACP}(g^{ACP}(X) = X^0 + U_K g^{ACP}(X) = X^0 + U_K U_K^T(X - X^0) \tag{2.27}$$

Soit, en notant $P_{U_K^\perp} = \mathbf{I}_K - U_K U_K^T$ (matrice de projection sur H_U) :

$$\begin{array}{l} \mathbb{E}||X - f^{ACP}(g^{ACP}(X))||^2_{R^J} \\ = \mathbb{E}||P_{U_K^\perp}(X - \mathbb{E}(X))||^2_{R^J} + \mathbb{E}||P_{U_K^\perp}(X^0 - \mathbb{E}(X))||^2_{R^J} \end{array} \tag{2.28}$$

On en déduit que $X^0 = \mathbb{E}(X)$ convient en annulant le deuxième terme positif, puis en réécrivant, l'expression devient :

$$\begin{array}{l} \mathbb{E}||X - f^{ACP}(g^{ACP}(X))||^2_{R^J} \\ = \mathbb{E}||X - \mathbb{E}(X)||^2_{R^J} - \sum_k u_k^T \mathbb{E}((X - \mathbb{E}(X))(X - \mathbb{E}(X))^T) u_k \end{array} \tag{2.29}$$

Ce qui indique que minimiser la dernière expression par rapport à U_K conduit à des vecteurs propres u_k diagonalisant la matrice de variance-covariance des données, et de plus grandes valeurs propres (cf. SVD). Par ailleurs, le critère de projection de l'ACP se ramène à celui des surfaces principales : il s'agit de projeter une donnée X sur H_U, et donc de minimiser la distance entre X et son projeté. En effet, $g^{ACP}(x_i) = argmin_{\xi \in \mathcal{R}^K} ||x_i - (\bar{X}_I + U_K \xi)||^2$. Enfin, il suffit de remplacer l'opérateur $\mathbb{E}$ par une somme sur un échantillon pour retrouver le cas d'un ensemble de données fini. □

2.3.2 Carte Auto-organisatrice (SOM) de Kohonen

Kohonen[38] a proposé dans le début des années 1980 la méthode des cartes auto-organisatrices[10] ou SOM pour *Self-Organizing Maps*.

2.3.2.1 Algorithmes du SOM

L'algorithme SOM est une variante de l'algorithme des K-means qui, lors d'une itération, modifie non seulement un centre sélectionné comme étant le plus proche d'une donnée, mais aussi les centres voisins pour un graphe de voisinage fixé. Le graphe implique des interactions latérales entre centres qualifiés de voisins. On définit une grille de K points $\xi_k \in \mathcal{R}^2$, indicés pour k allant de 1 à K, formant l'ensemble $\Xi = \{\xi_1, \xi_2, \cdots, \xi_K\}$. Contrairement aux Surfaces Principales, Ξ n'est plus une partie de $\mathcal{R}^2$ comme un ouvert ou un intervalle, mais un ensemble discret de points posés sur un plan. On note les valeurs latentes m_k correspondant à la valeur de f en ξ_k : les centres de classe s'écrivent $m_k^n = f^n(\xi_k)$ pour la n-ième itération. On obtient les nouvelles valeurs des centres à la manière d'un gradient stochastique. L'algorithme en ligne ou séquentiel du SOM (en supposant un coefficient α_n factorisé dans h^n (la fonction de voisinage à l'itération n), et décroissant avec le temps par $\eta \in]0, 1[$ ici) s'écrit :

[10] Vois la section consacré en fin du chapitre.

Algorithme 5 ***Algorithme en-ligne du SOM :***

SOM stochastique

1 *Initialisation de* θ^0, σ^0
2 *Sélection d'un individu* $x_{i_n} \in \mathcal{D}$,
3 *Calcul de la classe la plus proche*
$g^{n+1}(x_{i_n}) = argmin_{k \in \mathcal{K}} ||x_{i_n} - f^n(\xi_k)||_{\mathcal{R}^J}$,
4 *Recalcul des centres* ($\sigma^{n+1} = \eta\sigma^n$, *(après 1 cycle))*
$m_k^{n+1} = m_k^n + h_{k\, g^{n+1}(x_{i_n})}^{n+1} (m_k^n - x_{i_n})$,
5 $||\theta^n - \theta^{n-1}|| < \epsilon$ *alors* $\hat{\theta} = \theta^n$, *sinon retour à 2.*

Il s'agit d'un K-means séquentiel où les centres de classes sont recalculés itérativement sur l'ensemble des classes en introduisant une pondération k_{kl} décroissant pour une distance croissante sur le graphe de voisinage posé a priori. Si on suppose être proche de la solution[11], on aboutit à l'algorithme suivant :

Algorithme 6 ***Algorithme hors-ligne du SOM :***

SOM batch

1 *Initialisation de* θ^0, σ^0
2 *Recherche des centres les plus proches*
$g^{n+1}(X) = argmin_{k \in \mathcal{K}} ||X - f^n(\xi_k)||_{\mathcal{R}^J}$,
3 *Recalcul des centres* ($\sigma^{n+1} = \eta\sigma^n$),
$m_k^{n+1} = f^{n+1}(\xi_k) = \mathbb{E}[X | g^{n+1}(X) \in \mathcal{N}_k^{n+1}]$ $\forall k, n$,
4 *Test d'arrêt*
$||\theta^n - \theta^{n-1}|| < \epsilon$ *alors* $\hat{\theta} = \theta^n$, *sinon retour à 2.*

On a repris la notation dans l'article [37], pour la version batch : dans le cas d'un échantillon fini, il faut remplacer l'espérance par une moyenne empirique; $\mathcal{N}_k$ est l'ensemble des centres voisins de m_k. A la convergence ($n = \infty$), le voisinage disparaît se réduisant au neurone gagnant (projection discrète) et l'on retrouve la définition d'un K-means, avec convergence en un minimum local du fait de la contrainte. Pour vérifier si ce minimum n'est pas trop éloigné d'une solution plus optimale, il suffit de comparer la valeur du critère de variance-intra obtenu avec celui d'un K-means classique sans contrainte. Etant données les difficultés dans l'estimation et l'étude des propriétés

[11] En supposant un état stationnaire, on a $\forall k\ \mathbb{E}_n[m_k^{n+1} - m_k^n] = \mathbb{E}_n[h_{k,z_{i_n}}(m_k^* - x_{i_n})] = 0$, où $m_k^* = lim_{n\to\infty} m_k^n$ est la solution stationnaire du centre de C_k, et z_{i_n}, l'indice de la classe la plus proche de x_{i_n}. Ensuite, étant donné un échantillon fini de données, on en déduit directement la solution qui dépend du paramètre inconnu $\mathbf{m}^*$ en raison des z_{i_n}. Une résolution, en remarquant une forme de fonction contractante, donne finalement le SOM batch. Cette justification donnée ici brièvement est développée dans l'article de Kohonen[39]

du SOM, des auteurs ont proposé des alternatives sous la forme de modèles probabilistes paramétriques munis chacun d'un critère précis. Ils feront l'objet de la suite de ce chapitre.

2.3.2.2 Critère du SOM

Dans le SOM, le formalisme d'une fonction de voisinage est apparu peu de temps après le premier algorithme de Kohonen, la version originelle modifiant les centres dans un certain voisinage, sans lissage par une fonction décroissante avec la distance au plus proche centre. L'intérêt du SOM réside dans sa capacité à représenter sur une carte la distribution des données, au prix d'une certaine déformation ou distorsion. L'algorithme d'estimation originel est de nature stochastique[18]. Kohonen en dérive un second algorithme, hors-ligne ou déterministe obtenu en approchant la solution locale limite. Apparenté aux surfaces principales[40], le SOM peut être qualifié d'ACP non-linéaire (non-paramétrique, discrétisée). Les propriétés[12] de convergence et d'auto-organisation sont introduites sans démonstration par Kohonen[38] et sont seulement observées, comme indiqué par l'auteur lui-même, en pratique. Certains résultats de convergence existent[41, 42]. Le SOM, ne minimise[43] pas à proprement parler de critère, mais seulement de manière approchée, en négligeant les bords du pavage de Voronoï, et pour un échantillon fini, où $P(x_i)$ est la probabilité de l'échantillon x_i :

$$E_{\mathcal{SOM}}(\theta, \mathbf{C}|\mathcal{D}) = \sum_i P(x_i) \sum_k c_{ik} \sum_l h_{kl} ||x_i - m_l||^2_{\mathcal{R}^J} \tag{2.30}$$

En effet, une optimisation par gradient stochastique aboutit à l'algorithme original du SOM, et une maximisation directe donne le SOM batch (à la distance au centre près qui devient ici pondérée avec h, faisant intervenir le voisinage localement). Pour un échantillon infini, par contre, la somme devient une intégrale, avec des bords difficilement négligeables. D'ailleurs, Kohonen a obtenu une règle[44] de modification des centres de classes différentes, plus performante apparemment, mais sans fondement biologique, selon l'auteur. D'un point de vue analyse des données, nous supposons disposer d'un échantillon de données fini, donc l'existence d'un critère optimisé $E_{\mathcal{SOM}}(\theta, \mathbf{C}|\mathcal{D})$ est assuré.

2.3.2.3 Le voisinage du SOM

Les cartes de Kohonen construisent des représentations bidimensionnelles d'une distribution multidimensionnelle. La projection s'effectue donc au prix d'une certaine déformation quantifiée par le critère approché pour un échantillon fini.

Topologie de la grille

[12] A notre connaissance, il n'a été démontré rigoureusement ni la convergence ni l'auto-organisation de le carte en dimension supérieure à un et pour tous les cas

Nous avons présenté les cartes auto-organisatrices dans une optique d'analyse de données par projection classifiante sur le plan ; c'est pourquoi, la forme de grille étudiée est un treillis rectangulaire. Pourtant, l'algorithme est indépendant de la topologie de la grille. Une forme en ficelle, réduite à un segment de classes est par exemple envisageable ; elle s'apparente aux courbes principales. Un treillis non plus rectangulaire, mais hexagonal est envisageable et apparemment préférable pour diminuer les déformations de la projection. Une vue tridimensionnelle fournit un autre point de vue.

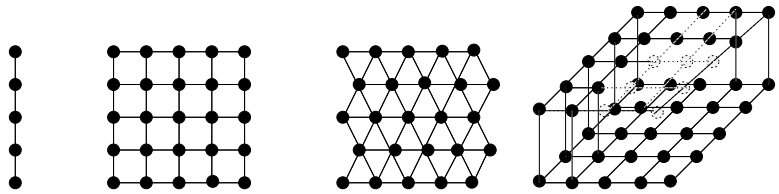


FIG. 2.2 – Exemples de topologies fréquentes de grille : Ficelle 1D, Rectangulaire 2D, Hexagonale 2D, Régulière 3D.

En fait, un utilisateur d'un système d'analyse de données basé sur le SOM devrait idéalement pouvoir choisir la configuration de classe, ou bien celle-ci devrait éventuellement s'adapter automatiquement à la distribution. On parle de *Growing Neural Gas*[45], notamment pour une quantification avec topologie de classes adaptative, avec liens dynamiques entre classes. Une méthode plus récente permet de faire croître également sur le plan le nombre de classes, avec un critère qui privilégie les zones moins bien représentées. Une croissance hiérarchique[46] plus ou moins maîtrisée, pour la grille est également envisageable. Il existe également d'autres formes variées de topologie avec des géométries diverses, ou même dans des espaces non euclidiens comme une version de carte hyperbolique[47]. Enfin, une variante pour la projection planaire d'un graphe, où les données s'identifient au graphe de voisinage, a été proposée[48].

Fonction de voisinage

L'algorithme d'estimation d'une carte de Kohonen emploie une fonction de voisinage pour lisser la moyenne des données à l'aide des données des centres voisins sur la carte. La fonction de voisinage, notée $h(k,l)$ calculée en (ξ_k, ξ_l) sur la grille, vérifie la propriété générale :

$$h(k,l_1) \leq h(k,l_1) \text{ si } |\xi_k - \xi_{l_1}||_{\mathcal{R}^2} \geq ||\xi_k - \xi_{l_1}||_{\mathcal{R}^2} \tag{2.31}$$

Il existe une variante de forme en cloche (que l'on retrouve par des contraintes bayésiennes) et munie d'un minimum local circulaire autour du maximum placé au centre ξ_k. Pour faire décroître le rayon d'activité de la fonction avec le temps, un

paramètre assimilable à une variance notée σ est diminué au cours des pas d'estimation. Par exemple, on peut employer une fonction bulle ou gaussienne, en indiçant explicitement σ (fonction du pas n) :

Fonction bulle	Fonction gaussienne
$h_\sigma(k,l) = \begin{cases} 1 & \text{si } \|\|\xi_k - \xi_l\|\|_{\mathcal{R}^2} \leq \sigma \\ 0 & \text{sinon} \end{cases}$	$h_\sigma(k,l) \propto \exp(-\frac{1}{2\sigma^2}\|\|\xi_k - \xi_l\|\|^2_{\mathcal{R}^2})$

Ci-contre on trouve une illustration de ce genre de fonction, pour le cas de la gaussienne, en $\xi_k = (2,5)$ fixé, le noeud où h atteint son maximum.

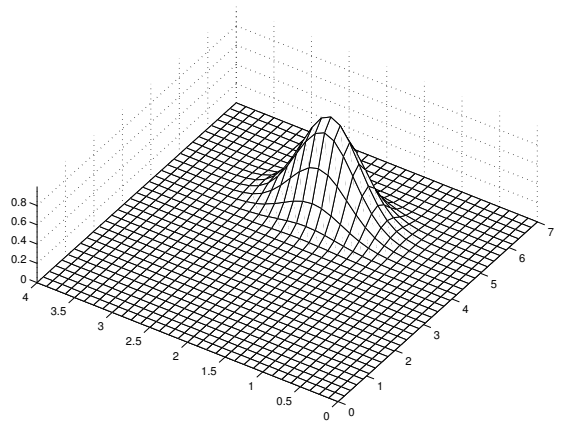


FIG. 2.3 – Exemple d'une fonction de voisinage gaussienne, grille 7*4.

En outre, factorisé dans l'expression de la fonction de voisinage, on doit en général considérer un coefficient décroissant avec le temps : α_n. En analogie avec le gradient stochastique, il faut que $\sum_n \alpha_n = \infty$ et $\sum_n \alpha_n^2 < \infty$ pour assurer la convergence. Cela est nécessaire également pour le K-means final lorsque la fonction de voisinage est réduite à la fonction delta de Kronecker[13] $\delta(k,l)$: l'affectation au centre le plus proche. En pratique, on utilise souvent une des formes variées suivantes pour A et B appropriés :

linéaire	rationnelle	exponentielle
$\alpha^n = A\,n + B$	$\alpha^n = \frac{1}{A\,n + B}$	$\alpha^n = A^n\,\sigma^0$

Cela permet finalement de définir la matrice $H \in \mathcal{R}^{K \times K}$ symétrique et de composantes h_{kl}. On indicera éventuellement par le temps n. Les méthodes probabilistes vont

[13] $\delta(k,l) = \begin{cases} 1 & \text{si } k = l \\ 0 & \text{sinon} \end{cases}$.

permettrent de formaliser soit la notion de voisinage, soit la notion de bruit dans les classes.

2.4 Surfaces Principales et mélanges de lois contraintes

On regroupe sous cette classe de méthodes les modèles probabilistes à variables latentes directement dérivés du SOM ou ACP. Par exemple le GTM[49] est une méthode récente qui tout comme le SOM permet d'obtenir la projection d'une distribution sur une surface (2D). La méthode est une variante non-linéaire d'une autre méthode appelée PPCA pour ACP probabiliste. On note généralement dans la suite une variable Z qualifiée de latente ou cachée ; le vecteur aléatoire des données sera nommé X ou X_i, de réalisations x ou x_i (éventuellement non différentié pour alléger la notation) ; il sera alors exprimé une forme paramétrique $P(x|z,\theta)$ ou $P(x_i|z,\theta)$ où θ est le vecteur des paramètres (pas toujours écrit dans des cas non ambigus) de la loi conditionnelle précédente, ainsi qu'une forme de la loi a priori de Z, P_Z, continue (gaussienne) ou discrète (somme de Diracs). Le critère d'estimation du vecteur de paramètre θ inconnu est un maximum de vraisemblance, classiquement estimé par la méthode numérique EM. Ainsi, on marginalise[14] sur Z sachant que le cas discret transforme l'intégrale en somme :

$$\forall x_i \ , \ P(x_i|\theta) = \int_\Omega P(x_i|Z,\theta) \ dP_Z \ \text{ , avec } \ \Omega = \mathcal{R}^2, \ \mathcal{Z} = \{1,2,\cdots,K\} \text{ ou } \mathcal{R}^J \quad (2.32)$$

D'où la logvraisemblance générale des modèles étudiés :

$$\mathcal{L}(\theta|\mathcal{D}) = \sum_i \log \int_\Omega P(x_i|Z,\theta) \ dP_Z \quad (2.33)$$

L'intérêt de la variable latente est d'introduire des indépendances stochastiques conditionnellement à cette variable, simplifiant quelque peu les calculs analytiques, mais sans pour autant perdre des corrélations (ou liaisons) éventuelles entre les composantes vectorielles puisque que l'on rappelle que cela n'implique pas obligatoirement l'indépendance par marginalisation. En fait, il ne s'agit plus de la variable aléatoire représentant un label de classe, mais plutôt un centre de classe contraint par la probabilité a priori $P(Z)$.

2.4.1 ACP probabiliste (PPCA)

L'ACP probabiliste[50] est présentée par Bishop et Tipping (et indépendamment par Roweis[51]) qui étendent des travaux précédents sur une méthode appelée Analyse en Facteurs communs spécifiques (*Factor Analysis*) issue des modèles à variable latente.

[14] Une écriture plus formelle est : $P(X = x_i|\theta) = \int_\Omega P(X = x_i|dz,\theta) dF_Z(dz)$ avec F_Z est la fonction de répartition de Z, et x_i une observation, et d_z l'élément infinitésimal d'intégration, et Ω l'espace probabilisé.

Ils montrent comment l'ACP peut être formulée[15] comme une solution du maximum de vraisemblance. Les résultats des auteurs reprennent ceux de travaux antérieurs sur l'ACP, en apportant des compléments théoriques et pratiques. Soit une variable latente Z telle que $Z \sim \mathcal{N}_J(0, I_L)$ où I_L est la matrice identité L-dimensionnelle. Une variable observée $x \in \mathcal{R}^J$ est alors définie comme une transformation de $z \in \mathcal{R}^L$ (réalisation de Z) avec un bruit additif gaussien : $X = WZ + \mu + \epsilon$ où $W \in \mathcal{R}^{J \times L}$, $\mu \in \mathcal{R}^J$ et $\epsilon \sim \mathcal{N}_J(0, \sigma^2 I_J)$:

Définition 16 *Le modèle du PPCA ou ACP probabiliste se définit par :*

$$\begin{array}{ll} y(z, W) = Wz & z \sim \mathcal{N}_L(0, I_L) \\ x = y(z, W) + \mu + \epsilon & \epsilon \sim \mathcal{N}_J(0, \sigma^2 I_J) \\ \theta = (W, \mu, \sigma) & \end{array} \tag{2.34}$$

On en déduit la densité de probabilité $P(x|z; W, \mu, \tau) = \mathcal{N}_J(Wz + \mu, \sigma^2 I_J)$, et finalement, on calcule la distribution de X par marginalisation sur Z, d'où la loi :

$$P(x|W, \mu, \sigma) = \int P(x|Z; W, \mu, \sigma) dP_Z \sim \mathcal{N}_J(\mu, \sigma^2 I_J + WW^T) \tag{2.35}$$

Cela définit une distribution gaussienne contrainte dépendant des paramètres W, μ et σ ; ces derniers peuvent s'estimer par maximum de vraisemblance de

$$\mathcal{L}(W, \mu, \sigma|\mathcal{D}) = \sum_i \log P(x_i|W, \mu, \sigma) \tag{2.36}$$

Les points stationnaires calculés par Bishop et Tipping vérifient :

$$\hat{W} = U_J(\Lambda_J - \sigma^2 I_J)^{1/2} R \quad \text{et, } \hat{\sigma}^2 = \frac{1}{J - L} \sum_{k=L+1}^{J} \lambda_k \tag{2.37}$$

où Λ_J et U_J sont respectivement les matrices de valeurs propres et de rotation associées à l'ACP de la matrice empirique de variance-covariance des $\{x_i\}_{i=1}^{i=I}$, et R une matrice quelconque de rotation. On retrouve l'ACP en faisant tendre σ vers zéro.

On peut représenter sur le plan les projections comme la moyenne sur z conditionnellement à la donnée x_i, avec $\hat{\mu}$ la moyenne empirique de l'échantillon :

$$\hat{E}_Z(Z|x_i) = \int ZP(Z|x_i, \hat{W}, \hat{\sigma}^2) dP_Z = (\hat{\sigma}^2 I_L + \hat{W}^T \hat{W})^{-1} \hat{W}^T (x_i - \hat{\mu}) \tag{2.38}$$

Ce modèle interprétant l'ACP comme un modèle probabiliste a permis dans un second temps l'apparition de méthodes bayésiennes de choix[29] du nombre de valeurs propres ou même de mélange[50] de PPCA qui modélise des ACPs [16] dans des classes, ainsi que l'algorithme du GTM (*Generative Topographic Mapping*).

[15] Comme l'ACP a été introduite originalement, avec la notation $\mathcal{N}(m, \Sigma)$, la loi de probabilité normale (gaussienne) multidimensionnelle d'espérance m et de matrice de variance-covariance Σ.

[16] Il est clair qu'une version type AFC probabiliste directement dérivée du modèle précédent est envisageable de manière très simple puisque l'on rappelle que l'AFC peut s'exprimer simplement comme une ACP. On en dériverait un algorithme de mélange fonctionnant pour un tableau de contingence.

2.4.2 Generative Topographic Mapping (GTM)

Le GTM[49, 52, 53] est une méthode non-linéaire récente à variable latente. Il modélise une grille de variables latentes équiprobables plongées dans un espace 2D ou 3D ; cette grille est projetée de manière non linéaire dans l'espace des données (continues). On peut considérer le GTM comme une variante[17] Monte-Carlo du PPCA. Dans le cas étudié ici, par grille, les projections des points de cette dernière sont les centres de gaussiennes isotropiques sphériques. Les données sont alors supposées provenir de ce mélange contraint de gaussiennes. L'algorithme doit trouver les paramètres optimaux de la fonction de projection ainsi que la variance commune des gaussiennes en maximisant la vraisemblance.

Plus formellement, $y(z, W)$ représente la projection non linéaire de Z, variable latente ($z \in \mathcal{R}^2$), treillis de distributions de Dirac équidistribuées sur une grille régulière. En fait, on pose M, le nombre de bases non-linéaires (gaussiennes ou autre) de la transformation Φ (semblable pour la régression linéaire généralisée) avec $J > M > L$, et :

$$W \in \mathcal{R}^{J\times M}\ ,\ \Phi(z) \in \mathcal{R}^{L\times M} \text{ t.q. } y(z,W) = W\Phi(z)\ ,\ \text{où } \Phi(z) = (\phi_1(z), \cdots, \phi_M(z))^T \quad (2.39)$$

Définition 17 *Le modèle du GTM ou Generative Topographic Mapping se définit par :*

$$\begin{array}{ll} y(z,W) = W\Phi(z) & z \sim K^{-1}\sum_k \delta(z-\xi_k) \\ x = y(z,W) + \mu + \epsilon & \epsilon \sim \mathcal{N}(0, \sigma^2 I_J) \\ \theta = (W, \mu, \sigma) & \end{array} \quad (2.40)$$

La logvraisemblance de l'échantillon s'écrit alors :

$$\mathcal{L}(\theta|\mathcal{D}) = \sum_i \log\left(\frac{1}{K}\sum_k^K P(x_i|\xi_k, W, \sigma)\right) \quad (2.41)$$

La maximisation d'une telle vraisemblance est relativement lourde. Les auteurs proposent une résolution par algorithme EM qui met en oeuvre une inversion de matrice. On note $\Phi = [\Phi(\xi_1)|\Phi(\xi_2)|\cdots|\Phi(\xi_K)]^T \in \mathcal{R}^{K\times M}$, D la matrice des données.

Algorithme 7 *On déduit de l'EM pour le GTM :*

Algorithme GTM

Init	*Initialisation de* θ^0,
Pas E	$R^n = [R^n_{ki} = P(\xi_k\|x_i, W^n, \sigma^n)] \in \mathcal{R}^{K\times I}$,
	$G^n = diag(\sum_i P(\xi_k\|x_i, W^n, \sigma^n)) \in \mathcal{R}^{K\times K}$,
Pas M	$(\Phi^n)^T G^n \Phi^n (W^{n+1})^T = (\Phi^n)^T R^n D$,
	$(\sigma^{n+1})^2 = \sum_i \sum_k p(\xi_k\|x_i, W^n, \sigma^n)\|\|W^{n+1}\Phi(\xi_k) - x_i\|\|^2$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
	sinon retour au Pas Estimation.

[17] En fait, il faudrait échantillonner la variable latente exactement suivant une gaussienne au lieu de prendre des Diracs sur une grille, ce qui approche plutôt une distribution uniforme.

De même que pour le PPCA, on peut représenter sur le plan la position moyenne d'une projection :

$$\int Z\, P(Z|x_i, \hat{W}, \hat{\sigma})\, dP_X = \sum_k^K \xi_k\, P(\xi_k|x_i, \hat{W}, \hat{\sigma}) \tag{2.42}$$

On pourra préférer une représentation par maximum du mode qui affecte à chaque individu le point de grille le plus probable. Cette représentation graphique est similaire au SOM :

$$\hat{z}_i = argmax_{\xi_k}\ P(\xi_k|x_i, \hat{W}, \hat{\sigma}) \tag{2.43}$$

Enfin, récemment, une version pour variable binaire est définie brièvement sans développement explicite par les mêmes auteurs. Là encore, la maximisation de la vraisemblance obtenue pose problème. Mark Girolami a développé une méthode améliorée dans son article [54]. Tipping [55] a aussi développé la visualisation de données binaires de grande dimension ; sa version non-linéaire utilise une variable latente gaussienne comme le fait le PPCA. Le modèle présenté ci-après est la première formalisation des courbes principales à l'aide des modèles de mélange. Le GTM demeure aujourd'hui, malgré ses limites pratiques, une méthode souvent considérée préférable au SOM, car beaucoup plus robuste. Dans la suite, nous introduisons brièvement d'autres approches bien moins employées en pratique dans la littérature.

2.4.3 Modèle de Tibshirani : Courbes Principales Revisitées (RPC)

L'article de Tibshirani[56], est le premier à notre connaissance à proposer une modélisation à base de lois des courbes principales. Il n'étudie que le cas à une dimension mais comme il le dit lui-même, l'extension à 2 dimensions est possible.
Le modèle que l'auteur décrit est le suivant. On suppose les x_i, réalisations i.i.d. multivariées dans $\mathcal{R}^J$ du vecteur aléatoire X de densité P_X. On imagine également que chaque x_i est généré en deux étapes : 1)une variable latente Z_i est générée selon une certaine distribution P_Z donnant la réalisation z_i, 2) x_i est générée à partir de la distribution conditionnelle $P_{X|Z}$ ayant pour moyenne $f(Z)$, un point d'une courbe dans $\mathcal{R}^J$, avec les composantes du vecteur X indépendantes conditionnellement à Z. La définition *revisitée* de courbe principale par Tibshirani s'écrit :

Définition 18 *Une courbe principale paramétrique est le triplet* $(P_Z, P_{X|Z}, f)$ *avec* $P_X \sim \int P_{X|Z} dP_Z$ *et* f *est une courbe paramétrée par* $z \in [z_{min}, z_{max}] \subset \mathcal{R}^1$, *avec* $f(z) = E(X|Z = z)$, *implicite. Avec* $P_{X|Z=z}$ *gaussien, on a :*

$$\begin{array}{l} x_i|z_i \sim \mathcal{N}(f(z_i), \Sigma(z_i)) \quad Z_i \sim P_Z \;\; f(Z) = E(X|Z) \\ \theta = (\{f(z)\}_{z\in[z_{min},z_{max}]}, \{\Sigma(z)\}_{z\in[z_{min},z_{max}]}) \end{array} \tag{2.44}$$

D'où la logvraisemblance :

$$\mathcal{L}(\theta|\mathcal{D}) = \sum_i \log \int P(x_i|Z,\theta) dP_Z \tag{2.45}$$

L'auteur justifie alors l'utilisation de points supports qui permettent de discrétiser la courbe en I points ξ_k et résout le problème par EM. Il obtient alors un algorithme comparable à l'estimation d'un modèle de mélange où les moyennes sont les points de la courbe aux points supports.

La définition de Hastie donne un algorithme optimisant, avec des splines f_j :

$$\sum_i \sum_j (x_{ij} - f_j(z_i))^2 + \sum_j \lambda_j \int_0^1 [f_j(z)^{(2)}]^2 dz \tag{2.46}$$

L'auteur propose également d'ajouter un tel lissage à l'aide des dérivées secondes, en contraignant les projections. Le modèle d'Utsugi exposé ci-dessous est une extension bayésienne de l'approche de Tibshirani, exempt de la question technique des points supports.

2.4.4 Modèle bayésien (GDM) de Utsugi (BSOM)

Utsugi[57, 58, 59], utilise un modèle de mélange de gaussiennes dont il contraint les centres à l'aide d'une loi a priori lissante gaussienne contenant un opérateur de différentiation laplacien. Il estime également les hyperparamètres du modèle à l'aide notamment d'une méthode empirique bayésienne couplée à une approximation normale, i.e. il approche l'intégration prise sur l'espace des centres de classe à l'aide d'une approximation gaussienne basée sur l'estimation de la hessienne pour obtenir une évidence approchée qu'il maximise. En outre, il montre que la maximisation par EM de son modèle aboutit à un algorithme effectivement proche du SOM.

Ici, les vecteurs x_i sont supposés multivariés dans $\mathcal{R}^J$. On note également $m(j) = (m_{1j}, m_{2j}, \cdots, m_{Kj})$ et $\alpha, \beta \in \mathcal{R}^+$. Ici, Z est la variable aléatoire de classe prenant ses valeurs dans $\mathcal{Z}$, et de réalisations Z_i correspondant aux valeurs inconnues des Z_i v.a. distribuée comme Z pour chaque X_i. On suppose L l'opérateur différentiel tel qu'un laplacien discrétisé[18].

[18]Par exemple, on prend pour L un opérateur matriciel de différentiation discrétisé sur l'espace topologique, tel que, en 1-D :

$$l_{ij} = \begin{cases} 0 & i+1=j \\ 1 & i+1=j\pm 1 \\ -2 & sinon \end{cases} \tag{2.47}$$

En outre, on note, puisque L peut être singulière, det^+, la déterminant pris comme produit des valeurs propres positives. La rang de $L^T L$ est l'entier l. En 2D, l'opérateur de différentiation est plus complexe.

Définition 19 *Le modèle d'Utsugi peut s'écrire, avec noté* **m**, *la variable aléatoire ou bien la réalisation des centres des gaussiennes, contraints par une loi a priori, permettant de disposer les centres sur une surface régulière :*

$$\begin{array}{|l} x_i \sim \sum_k \frac{1}{K}\mathcal{N}(m_k, \frac{1}{\beta}I_J) \quad m(j) \sim \mathcal{N}(0, (\alpha L L^T)^{-1}) \\ \theta = (\{m_k\}_k, \alpha, \beta) \end{array} \tag{2.48}$$

On l'estime par EM en marginalisant et en calculant la fonction $\mathcal{Q}$. Le lecteur intéressé par leur estimation trouvera une solution dans l'article d'origine[58], à l'aide d'une méthode bayésienne approchée. Ici, α et β sont supposés connus pour simplifier la présentation. On obtient l'algorithme, avec $\gamma = \alpha/\beta$,

BSOM d'Utsugi

Init	Initialisation de θ^0,
Pas E	$t_k^n(x_i) = P(x_i \vert m_k^n, \beta) / \sum_l P(x_i \vert m_l^n, \beta)$,
Pas M	$N^n = diag(\bar{m}_{1j}^n, \bar{m}_{2j}^n, \cdots, \bar{m}_{Kj}^n)$,
	$\forall j,k\ \bar{m}_{kj}^n = \frac{1}{\sum_i t_k^n(x_i)} \sum_i t_k^n(x_i) m_{kj}^n$ et $m_{(j)}^{n+1} = (N^n + \gamma L^T L)^{-1} N^n \bar{m}_{(j)}^n$,
Test	$\vert\vert\theta^n - \theta^{n-1}\vert\vert < \epsilon$ alors $\hat{\theta} = \theta^n$,
	sinon retour au Pas Estimation.

Ce modèle inadapté au traitement de variables qualitatives sera étendu au cas multinomial dans le chapitre suivant. Dans la suite, nous décrivons des modèles employant soit une variable aléatoire non observée gaussienne répartie sur le plan (cf. PPCA), soit une grille de distribution de Dirac (cf. GTM) : ce sont des alternatives à un loi a priori par Laplacien sur les composantes des centres de classe. Nous décrivons essentiellement le cas de vecteurs binaires, même si le cas multinomial est également traité par Girolami.

2.4.5 Modèles dérivés pour le cas de variables binaires qualitatives

Girolami et Tipping introduisent tous les deux (indépendamment) un modèle de mélange de lois de Bernoulli multivariées avec une représentation logit des probabilités à partir d'une variable cachée distribuée ; dans la suite les données x_i sont donc des réalisations de v.a. discrètes binaires.

Plus formellement, on garde la même formulation du modèle général des modèles précédents ; par contre, les formes paramétriques des distributions considérées sont modifiées en introduisant la loi des composantes $P_i(v_j|z)$ spécialement adaptée :

$$P(x_i|\theta) = \int_{\mathcal{R}^2} P(x_i|Z;\theta) dP_Z = \int_{\mathcal{R}^2} [\prod_j^J P_i(v_j|Z)]\, dP_Z \tag{2.49}$$

On note[19] Z une variable latente définie de manière exacte plus loin. On pose B_{ij} les variables binaires renseignant sur la présence ou absence d'un mot v_j dans le texte x_i :

$$P(x_i|z;\theta) = \prod_j^J P_i(v_j|z) = \prod_j^J P(v_j|z)^{B_{ij}} (1 - P(v_j|z))^{1-B_{ij}} \tag{2.50}$$

où,

$$P(v_j|z) = \frac{exp(\beta_j^T z + b_j)}{1 + exp(\beta_j^T z + b_j)} \tag{2.51}$$

On note une telle transformation par GLM_{BM}, pour un modèle linéaire généralisé (GLM) avec modèle de bruit bernoullien multivarié (BM) :

$$\begin{array}{|l} x_i|z \sim GLM_{BM}(z,\beta) \quad z \sim ? \\ \theta = (\{\beta_j\}_j, \{b_j\}_j, ?) \end{array} \tag{2.52}$$

Enfin, Tipping considère la variable latente continue à distribution gaussienne comme en ACP probabiliste, tandis que Girolami considère une distribution discrète disposée en grille de deux dimensions, comme le GTM.

Pour résoudre la marginalisation, les deux auteurs emploient une approximation variationnelle du logit, proposé originalement par Jaakkola et Jordan dans un modèle bayésien logistique. Il s'agit d'une expression gaussienne à paramètres inconnus :

$$\tilde{P}_i(v_j|z,\epsilon_j) = S_j \, exp\{(R_{ij} - \epsilon_j)/2 + T_j(A_{ij}^2 - \epsilon_j^2)\} \tag{2.53}$$

avec $R_{ij} = (2B_{ij} - 1)(\beta_j^T z + b_j)$, $S_j = \frac{exp(\epsilon_j)}{1+exp(\epsilon_j)}$, et $T_j = (0.5 - S_j)/2\epsilon_j$. En effet, cette gaussienne (après estimation des paramètres) combinée à la distribution a priori de la variable latente s'intègre analytiquement, sachant que $\tilde{P}_i(v_j|z,\epsilon_j) \leq P_i(v_j|z)$.

Variable latente discrète (BGTM)

Pour l'approche de Girolami[54, 60], la variable latente Z prend ses valeurs dans $\{\xi_1, \xi_2, \cdots, \xi_K\}$ ou $\mathcal{Z}$ pour alléger la notation. On définit une grille de Diracs mais non équiprobables contrairement au GTM.

Définition 20 *Le modèle de Girolami s'écrit :*

$$\begin{array}{|l} x_i|z \sim GLM_{BM}(z,\beta) \quad z \sim \sum_k \pi_k \delta(z - \xi_k) \\ \theta = (\{\beta_j\}_j, \{b_j\}_j, \pi_k) \end{array} \tag{2.54}$$

[19] En réalité, Girolami propose dans l'exponentielle $\tilde{\beta}_j^T \phi(z)$, où ϕ transforme de façon non linéaire z, donc en prenant $\tilde{\beta}_j = (\beta_j, b_j)$ et $\phi(z) = (z, 1)$, on retombe sur le modèle de mot de Tipping.

Ce qui donne pour x_i :

$$P(x_i|\theta) = \int_{\mathcal{R}^2} P(x_i|Z,\theta)dP_Z = \sum_k^K \pi_k P(x_i|k;\theta) \tag{2.55}$$

On peut ensuite maximiser la vraisemblance par EM. La fonction $\mathcal{Q}$ s'écrit, en notant $t_k^n(x_i) = P(k|x_i,\theta^n)$:

$$\begin{array}{ll} Q(\theta|\theta^n) & = \sum_i \sum_j \sum_k t_k^n(x_i)\{B_{ij}\log P(v_j|k,\theta) \\ & +(1-B_{ij})\log(1-P(v_j|k,\theta)) + \log\pi_k\} \end{array} \tag{2.56}$$

La partie *Expectation* de l'EM consiste à calculer les $t_k^n(x_i)$ en fonction des $P(x_i|k;\theta)$. La partie *Maximisation* par contre nécessite une résolution par optimisation non linéaire. Par exemple une descente du gradient (GEM) comme le propose l'auteur, sachant que le gradient pour les β_j est (s pour abscisse et ordonnée) :

$$\frac{\partial Q}{\partial \beta_{sk}} = \sum_i \sum_k t_k^n(x_i)\{B_{ij} - P(v_j|\xi_k,\theta)\xi_{sk}\} \tag{2.57}$$

Les paramètres de mixages $P(\xi_k|\theta))$ se calculent directement par :

$$\pi_k^{n+1} = \frac{\sum_i t_k^n(x_i)}{I} \tag{2.58}$$

L'auteur donne une expression matricielle de la maximisation par méthode du gradient pour cette phase de l'EM. De même, il propose une méthode variationnelle alternative. Un des ces travaux propose également un solution GEM avec une expression matricielle élégante, adaptée à toute la famille de lois exponentielles.

Variable latente gaussienne (BPPCA)

Tipping[55] doit résoudre le problème tout autant analytiquement insoluble de maximisation de la logvraisemblance pour Z gaussienne :

Définition 21 *Le modèle de Tipping s'écrit :*

$$\begin{array}{|l} \hline x_i|z \sim GLM_{BM}(z,\beta) \quad z \sim \mathcal{N}(0,I_2) \\ \theta = (\{\beta_j\}_j, \{b_j\}_j) \\ \hline \end{array} \tag{2.59}$$

D'où :

$$\begin{array}{ll} P(x_i|\theta) & = \int_{\mathcal{R}^2} \prod_j^J P_i(v_j|Z,\theta)dP_Z \\ & = \frac{1}{2\pi}\int_{\mathcal{R}^2} \prod_j^J P_i(v_j|z,\theta)\ e^{-\frac{1}{2}||z||^2}dz \end{array} \tag{2.60}$$

Ce genre d'intégrale peut se calculer à l'aide de méthodes de type Monte-Carlo, ce qui a été proposé (d'après l'auteur) récemment dans la littérature, i.e. avec L grand et les ξ_k échantillonnés suivant la distribution gaussienne $P(Z)$:

$$\mathcal{L}(\theta|\mathcal{D}) = \sum_i \log \sum_k \prod_j P_i(v_j|\xi_k,\theta) - I\log K \tag{2.61}$$

Tipping use de l'approximation variationnelle du modèle logit. Il en déduit un algorithme performant qui se révèle plus rapide que la méthode Monte-Carlo en pratique. Une façon d'employer les modèles de mélange sans poser de loi a priori sur les classes est l'approche floue comme décrit ci-après.

2.4.6 Cartographie associative (TPEM)

Dans cette section, $x_i \in \mathcal{R}^J$ est à nouveau un vecteur multivarié. On décrit l'algorithme du *Topology Preserving EM* ou TPEM[61] pour une *cartographie associative*. On se référera à l'article d'origine pour le détail de la méthode. Le TPEM utilise un mélange de gaussiennes. Contrairement au SOM, les classes sont de matrices de variance-covariance pleines et différentes ; cette propriété lui évite de placer un plus grand nombre de centres dans les zones de forte densité. Cette propriété rend par contre l'estimation plus délicate en raison des paramètres plus nombreux. Le TPEM permet l'estimation et l'organisation en carte de distributions quelconques. Les auteurs ont justifié la méthode seulement pour un mélange de gaussiennes $\mathcal{N}(m_k, \Sigma_k)$ de moyennes m_k et matrices de variance-covariance Σ_k.

Définition 22 *La méthode TPEM optimise*[20] *la vraisemblance classifiante associée au modèle :*

$$\begin{array}{|ll|}\hline x_i|z_i = k \sim \mathcal{N}(m_k, \Sigma_k|\theta) & z_i \sim \mathcal{M}(1, (\pi_k)_k) \\ \theta = (\{m_k\}_k, \{\Sigma_k\}_k, \{\pi_k\}_k) & \\ \hline\end{array} \tag{2.62}$$

Le TPEM lisse les coefficients binaires c_{ik} en des valeurs μ_{ik}, après le pas d'affectation de classification CEM. Le lissage est tel qu'autour de la classe correspondante à l'étiquette k, la plus probable, de x_i, les coefficients d'affectation aux classes deviennent $\mu_{il} \propto h_{kl}$. La fonction de voisinage est diminuée en fonction du temps jusqu'à aboutir au delta de Kronecker en k.

On rappelle que dans le cas de gaussiennes isotropiques équiprobables, l'estimation du mélange classifieur et la minimisation du critère moyennes mobiles (ou critère de variance intra-classe) sont équivalentes. Les auteurs font alors le lien entre les K-means et le SOM car pour un échantillon fini, le SOM minimise approximativement le critère variance intra étendue (voir partie sur le SOM dans la section en fin de chapitre). Finalement, le TPEM est une variante d'EM classifiant CEM dont la phase d'affectation est rendue floue. La convergence est assurée car les derniers pas se transforment en un algorithme CEM puisque les coefficients d'affectation μ_{ik} deviennent binaires et s'identifient aux c_{ik}. Le modèle suivant formalise le voisinage ici flou en introduisant une seconde variable aléatoire non observée sur laquelle se distribue celle conditionnant les gaussiennes du mélange classifiant précédent.

[20] On rappelle que $\mathcal{M}$ est la loi de probabilité Multinomiale.

2.4.7 Carte topologique probabiliste (PRSOM)

Cette méthode consiste en une version probabiliste du SOM, à l'aide d'un mélange de mélanges de gaussiennes. Le modèle est dû à Luttrell[62] alors que l'estimation sous la forme des nuées dynamiques provient de travaux présentés à l'école Modulad[63]. On écrit pour $x_i \in \mathcal{R}^J$:

Définition 23 *Un modèle de SOM probabiliste s'écrit :*

$$\begin{array}{l} x_i \sim \sum_k \pi_k \sum_l \pi_{kl} P(x_i|a_l) \ \ où \ x_i|z_i = k \sim \mathcal{N}(m_k, \sigma_k) \\ \theta = (\{\pi_k\}_k, \{\pi_{kl}\}_{kl}, \{m_k\}_k, \{\sigma_k\}_k) \ \ et \ a_k = (m_k, \sigma_k) \end{array} \tag{2.63}$$

Un façon de résoudre le modèle pour un jeu de données i.i.d. est de considérer la vraisemblance classifiante et d'estimer à part les π_k. On aboutit à l'algorithme suivant :

Algorithme PRSOM

Init	Initialisation de θ^0,
Pas Affectation	$z_i^n \leftarrow argmin_k \sum_l \pi_{kl} P(x_i\|a_l^n)$,
Pas Maximisation	$m_k^{n+1} = \frac{\sum_i \pi_{kz_i^n} \frac{P(x_i\|a_k^n)}{P(x_i\|a_{z_i^n})} \ x_i}{\sum_i \pi_{kz_i^n} \frac{P(x_i\|a_k^n)}{P(x_i\|a_{z_i^n})}}$,
	$\sigma_k^{n+1} = \sqrt{\frac{\sum_i \pi_{kz_i^n} \frac{P(x_i\|a_k^n)}{P(x_i\|a_{z_i^n})} \ \|x_i - m_k^n\|^2_{\mathcal{R}^J}}{J \ \sum_i \pi_{kz_i^n} \frac{P(x_i\|a_k)}{P(x_i\|a_{z_i^n})}}}$,
Test Arrêt	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ alors $\hat{\theta} = \theta^n$,
	sinon retour au Pas Affectation
Fin	$\hat{\pi}_k = \frac{\sum_i \mathbb{1}_{\{k\}}(\hat{z}_i)}{I}$

On remarque avant tout que l'affectation n'est pas celle du SOM classique mais une version pondérée à l'aide de la fonction de voisinage, sur les probabilités d'un document dans une classe. Ensuite cette variante du SOM batch pondéré les x_i à l 'aide d'une fonction sensible à la probabilité dans la classe considérée. Enfin, il est possible de faire décroître π_{kl} en fonction du temps, comme pour le SOM. La même idée peut s'appliquer au mélange du PLSA, comme le fait Hofmann en proposant le TPLSA, méthode décrite ci-dessous.

2.4.8 PLSA topologique (TPLSA)

Cette méthode proposée par l'auteur du PLSA, dans [64], utilise un ensemble de facteurs fixes sur une grille en créant une distribution artificielle entre ces nouveaux facteurs et ceux du PLSA. Alors, par somme sur ces facteurs de la grille ou marginalisation, le nouvel algorithme favorise la répartition topologique. Plus formellement, on utilise les notations suivantes : Z' pour le facteur latent PLSA et Z'' pour la variable de

grille. L'idée principale est de fixer les probabilités $P(Z'|Z'')$ à l'aide d'une distance sur la grille, i.e., on considère chaque variable Z'_i et Z''_i correspondant de manière unique à une vecteur de coordonnées cartésiennes, d'où :

Définition 24 *Le PLSA topologique se définit à partir du PLSA en ajoutant une variable cachée distribuée artificiellement telle que* $P(Z' = k|Z'' = l) \propto h_{kl}$. *On suppose une distribution multinomiale des cellules de la matrice des données. Les probabilités des composantes suivent elles-mêmes un mélange discret distribuant la variable latente* Z' *du PLSA sur une autre variable artificielle* Z'' *:*

Distribution Multinomiales de D, v.a. des cellules d'un tableau de contingence

$$P_{j|i} = \sum_k P_{j|k} \sum_l P_{k|l} P_{l|i} \qquad (2.64)$$
$$\theta = (\{P_{j|k}\}_{j,k}, \{P_{l|i}\}_{l,i})$$

On résoud par EM, d'où l'algorithme :

Algorithme 8 *Le PLSA topologique itère jusqu'à convergence :*

Algorithme PLSA topologique

Init *Initialisation de* θ_0, σ_0,

Pas E *Voir PLSA,*
$P^n_{l|i,j} \propto P^n_{l|i} P^n_{j|l}$, *avec* $P^n_{j|l} = \sum_k P^n_{j|k} P_{k|l}$
$P^n_{k|i,j} \propto P^n_{k|i} P^n_{j|k}$, *avec* $P^n_{k|i} = \sum_v P^n_{v|i} P_{k|l}$

Pas M $\sigma^n = \eta \sigma^{n-1}$, $P_{k|l} \propto h^n_{kl}$
$P^{n+1}_{j|k} \propto \sum_i N_{ij} P^n_{k|i,j}$,
$P^{n+1}_{l|i} \propto \sum_j N_{ij} P^n_{l|i,j}$,

Test $||\theta^n - \theta^{n-1}|| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
sinon retour au Pas Estimation.

Après convergence, on obtient conditionnellement à y, la répartition des mots et documents, d'où la grille résumant l'information textuelle.

2.5 Conclusion

Ces méthodes bien qu'intéressantes dans le développement de leur modèle s'avèrent relativement lourdes en pratique ; elles sont peu adaptées au traitement des données textuelles et sont peu interprétables. D'ailleurs, la distribution multinomiale compétitive dans le domaine textuel n'y apparaît pas de façon efficace. C'est pourquoi, nous proposons dans la suite une nouvelle méthode tout spécialement développée pour l'analyse textuelle. De nouvelles propriétés des cartes auto-organisatrices s'en déduisent alors naturellement.

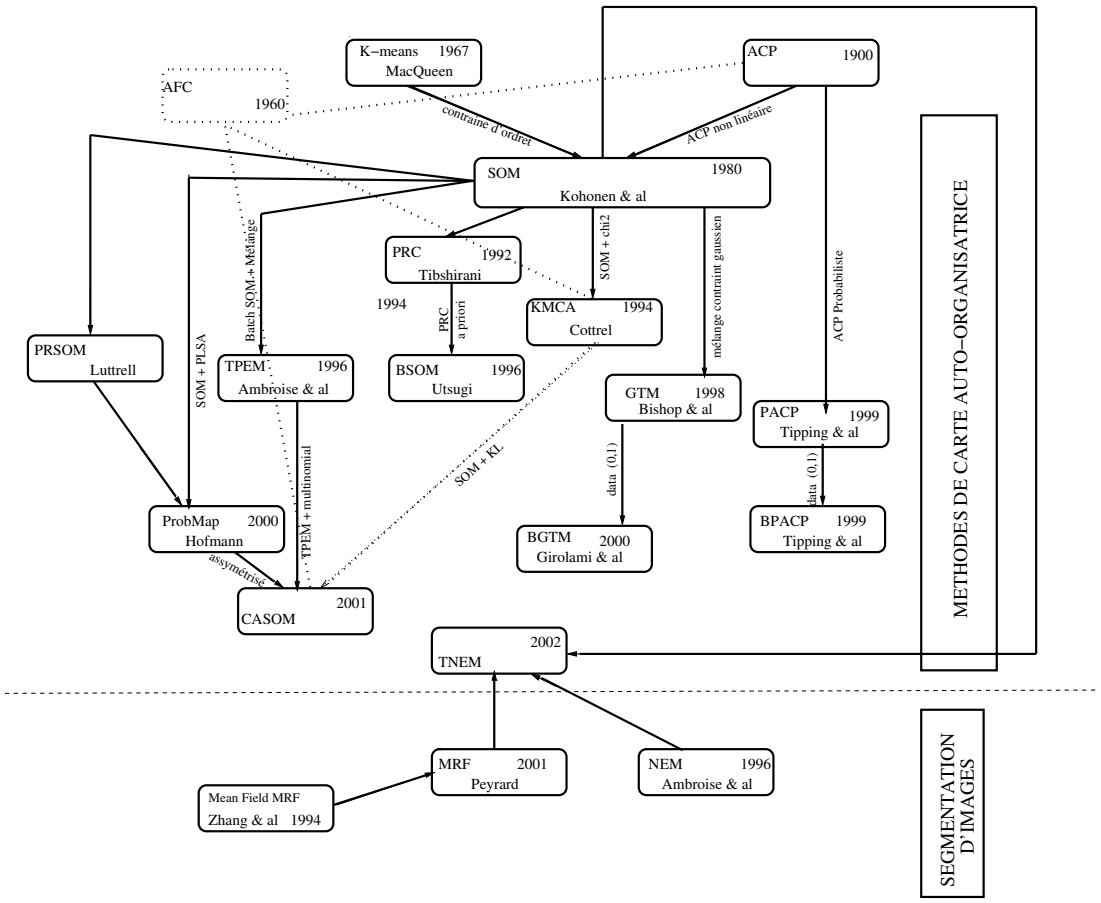


FIG. 2.4 – Historique des principales méthodes pour la projection sur une surface principale ainsi que de celles proposées dans le deuxième chapitre ; et illustration de l'analogie faite avec le traitement d'images.

Chapitre 3

Cartes auto-organisées pour les matrices de comptage

En analyse textuelle, la méthode d'analyse factorielle des correspondances donne des résultats synthétiques satisfaisants du point de vue qualitatif, comme en témoigne son usage courant dans les communications du domaine notamment linguistique. Nous recherchons une méthode alternative performante en cas de volume de données important, de type carte de Kohonen. Nous la voulons pourvue d'indicateurs statistiques apportant une aide à l'interprétation des résultats obtenus.

Ainsi nous adaptons le TPEM présenté dans le chapitre précédent afin de traiter un tableau de contingence. En effet, le principe de cette méthode, variante performante des cartes de Kohonen, est apte à projeter des vecteurs de distribution, comme des profils lignes d'un tableau de contingence : en remplaçant la distribution gaussienne de classe par une distribution multinomiale. On l'appelle CASOM pour SOM par *CA* (ou AFC). Ensuite, nous dégageons divers résultats à propos de CASOM : la distance approchée employée par la méthode, le comportement des critères qu'elle optimise. Nous nous intéressons également aux notions de qualité de la carte obtenue et plus particulièrement à la visualisation qui en dérive. Alors, nous proposons un nouveau formalisme du voisinage pour SOM en trouvant un critère flou à partir des modèles de la segmentation d'images. Cet algorithme illustre l'analogie que nous proposons entre les algorithmes des cartes auto-organisatrices et ceux de segmentation d'image. Puisque ce critère demeure encore flou, nous proposons un autre critère flou obtenu à partir d'une version bayésienne du SOM. Nous aboutissons alors à une version complètement probabiliste dont l'estimation est envisageable par une procédure d'EM généralisé.

3.1 CASOM : un SOM pour tableau de contingence

3.1.1 Algorithme CASOM

Etant donnés les bons résultats en classification textuelle obtenus par le mélange de lois multinomiales, nous supposons chacun des documents généré par une multinomiale particulière, posée comme la distribution de la classe à laquelle il appartient, et dont il conviendra d'estimer le vecteur de paramètre $P_{k\bullet} = (P_{1|k}P_{2|k}\cdots P_{J|k})$. L'algorithme du TPEM de [61] permet d'effectuer une cartographie associative à partir d'un mélange de gaussiennes. Aussi, nous appliquons le même principe pour notre modélisation adaptée à des données de comptage.

Définition 25 *Une méthode de carte auto-organisatrice adaptée aux tableaux de contingence est une adaptation du TPEM : on remplace la distribution gaussienne par une multinomiale ordonnée, d'où la logvraisemblance classifiante suivante :*

$$\mathcal{L}_{\mathcal{M}}(\theta, \mathbf{C}|\mathcal{D}, N_{1\bullet}, N_{2\bullet}, \cdots, N_{I\bullet}) = \sum_k \sum_i c_{ik} \log(\pi_k \prod_j P_{j|k}^{N_{ij}})$$

Ici, les paramètres c_{ik} qui permettent d'affecter une donnée à une et une seule classe, seront rendus flous en μ_{ik} à la façon du TPEM, pour induire une organisation en grille. Les marges du tableau apparaissent ici explicitement pour souligner qu'elles sont supposées fixées; dans la suite, elles ne sont pas notées pour alléger l'écriture.

Le modèle sous-jacent au TPEM traité par ses auteurs, Ambroise et Govaert, est le mélange particulier[61] de lois gaussiennes multidimensionnelles. Dans notre cas, nos données amènent naturellement à l'usage de la distribution multinomiale ici ordonnée $P(x_i|k, \theta^n) = \prod_j P_{j|k}^{N_{ij}}$. Notre algorithme se base donc sur le mélange de lois multinomiales. Du fait des marges intervenant dans la loi multinomiale, nous modélisons l'ensemble du tableau en considérant des marges constantes : un tel modèle peut d'ailleurs s'employer avec la distribution multinomiale classique. Finalement, ce modèle suppose données les marges du tableau de façon déterministe. L'algorithme d'estimation résultant est très semblable à l'estimation par EM du mélange de lois multinomiales, excepté le fait que les probabilités a posteriori sont remplacées par la forme lissée du SOM ou TPEM. Lors de l'estimation des centres multinomiaux du mélange contraint, un lissage bayésien classique est également introduit comme il est expliqué dans la section suivante, afin d'éviter des composantes vectorielles nulles difficilement manipulables. On a :

Algorithme 9 *L'algorithme proposé pour un tableau de contingence est une variante du TPEM. On répète*[1] *jusque convergence, avec une fonction de voisinage gaussien :*

[1] On initialise avec un σ_0 valant environ la taille du coté maximum de la grille. La répartition initiale des documents dans les classes est prise aléatoire.

Algorithme CASOM

Init	*Initialisation de* θ^0,
Pas Affectation	$\forall i\ z_i^n = argmax_{k \in \mathcal{Z}}\ \pi_k^n P(x_i \vert k, \theta^n)$,
Pas Paramètres	$\sigma^{n+1} = \eta \sigma^n$,
	$\forall i,k\ \mu_{ik}^n \propto h^n(\xi_k, \xi_{z_i^n})$,
Pas Maximisation	*Si* $\sum_i \mu_{ik}^n > 0$, $\forall j,k\ P_{j\vert k}^{n+1} = \frac{\sum_i \mu_{ik}^n N_{ij}+1}{\sum_i \mu_{ik}^n N_{i\bullet}+J}$,
	Sinon, $P_{k\bullet}^{n+1} = P_{k\bullet}^n$,
	$\forall k\ \pi_k^{n+1} = \frac{\sum_i \mu_{ik}^n+1}{\sum_i \sum_{k'} \mu_{ik'}^n+K}$,
Test	$\vert\vert\theta^n - \theta^{n-1}\vert\vert < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
	sinon retour au pas Affectation.

On a modifié un centre de classe si et seulement si des coefficients lissants non nuls lui sont affectés. Il arrive en effet que des classes soient vides, celles-ci servent alors de frontière. Sans cette condition supplémentaire au TPEM original, nous aboutirions pour les classes vides à des centres sans signification, aux composantes toutes identiques. On a noté η un coefficient dans]0,1], proche 1, permettant de diminuer exponentiellement le rayon de voisinage durant l'apprentissage, par exemple 0,9. La convergence est assurée puisque $h^n(\xi_k, \xi_{z_i^n}) \sim \delta(\xi_k, \xi_{z_i^n})$ pour n assez grand, et donc $\mu_{ik}^n \sim \delta(\xi_k, \xi_{z_i^n})$: l'algorithme se termine par des itérations d'EM classifiant. Nous étudions dans le paragraphe suivant le critère minimisé par l'algorithme ; pour ce faire, on se place après la phase d'organisation de la carte, pour le cas de μ_{ik} binaires.

3.1.2 A propos du lissage des multinomiales

Il est effectué un lissage[65] lors de l'estimation des probabilités multinomiales. En effet, l'algorithme rencontre fréquemment des groupes textes qui ne comportent pas certains mots du vocabulaire. Par conséquent, sans lissage les probabilités seraient nulles, ce qui occasionnerait des problèmes d'évaluation de la vraisemblance. En remarque, le lissage corrige ce problème tout en demeurant non biaisé asymptotiquement.

Le lissage est équivalent à l'introduction d'une loi a priori sur les probabilités multinomiales. En effet, pour une approche bayésienne de la distribution multinomiale, il est courant d'utiliser une distribution a priori de Dirichlet notée $Dir(\alpha_1, \alpha_2, \cdots, \alpha_J)$ sur les probabilités, telle que :

$$P(x_i, p \vert k, \theta, \alpha) = \underbrace{\prod_j P_{j\vert k}^{N_{ij}}}_{P(x_i \vert p,k,\theta)} \times \underbrace{\frac{\Gamma(\sum_j \alpha_j)}{\sum_j \Gamma(\alpha_j)} \prod_j P_{j\vert k}^{\alpha_j - 1}}_{P(p\vert k,\alpha) \sim Dir(\alpha_1,\alpha_2,\cdots,\alpha_J)} \propto \prod_j P_{j\vert k}^{N_{ij}+\alpha_j-1}$$

Le formalisme bayésien permet d'écrire, en sommant convenablement :

$$P(p \vert x_i, k, \alpha) \sim Dir(N_{ij} + \alpha_j) \text{ et } P(x_i \vert \alpha) = \frac{\Gamma(\sum_j \alpha_j)}{\sum_j \Gamma(\alpha_j)} \prod_j \frac{\Gamma(N_{ij} + \alpha_j)}{\Gamma(\alpha_j)}$$

Il est possible ici, d'un point de vue bayésien, de maximiser l'*evidence*, qui représente la probabilité d'avoir un échantillon provenant d'un distribution multinomiale. Ici, nous posons $\alpha_j = 1$ de façon a supposer une distribution uniforme. On en déduit la *posterior predictive distribution* :

$$P(v_j|x_i, k, \alpha) = E(P_{j|k}|x_i) = \frac{N_{ij} + 1}{N_{i\bullet} + J}$$

On peut choisir d'autres lois a priori. Par exemple pour $\alpha_j = 1/2$, on a l'a priori de Jeffrey.

3.1.3 Propriétés de la distance induite par le critère de vraisemblance

En réarrangeant la vraisemblance classifiante de multinomiales, on reconnaît deux sommes pondérées :

$$\begin{aligned}\mathcal{L}_{\mathcal{M}}(\theta, \mathbf{C}|\mathcal{D}) &= -N_{\bullet\bullet}\Big(\sum_i f_i \sum_k \mu_{ik}\ KL(f_{i\bullet}||P_{k\bullet}) - \frac{1}{N_{\bullet\bullet}}\sum_k\sum_i \mu_{ik}\ \log\pi_k\Big) \\ &+ N_{\bullet\bullet}\Big(\sum_i f_i \sum_k \mu_{ik} \sum_j f_{j|i}\log f_{j|i}\Big)\end{aligned}$$

La somme d'entropies est une constante puisqu'il s'agit des distributions empiriques. L'expression obtenue est très comparable au critère K-means puisqu'il suffit de remplacer la distance euclidienne par la divergence[2] KL et d'ajouter le facteur de pondération. Ce facteur de pondération f_i prend en compte la taille des textes. Contrairement aux K-means, un terme supplémentaire portant sur le nombre de classes induit une répartition des individus entre classes non uniforme : l'algorithme n'aura pas tendance à maintenir un nombre d'individus identiques dans l'ensemble des classes. Etant donné que ce terme fait intervenir l'inverse du total d'occurrences des mots dans le corpus, on le négligera dans la suite, lors de l'étude du critère.

Le maximum de vraisemblance ne minimise plus directement une variance-intra en terme de distance euclidienne entre des centres et un ensemble d'individus à classer. Il minimise la somme des divergences de Kullback entre les distributions empiriques $f_{i\bullet}$ des documents et les distributions de classe $P_{k\bullet}$ à calculer, pondérées par la probabilité empirique d'apparition de chacun des documents.

Définition 26

$$E_{KL}(\theta, \mu|\mathcal{D}) = \sum_i f_i \ \sum_k \ \mu_{ik}\ KL(f_{i\bullet}||P_{k\bullet}) - \tfrac{1}{N_{\bullet\bullet}}\sum_k\sum_i \mu_{ik}\ \log\pi_k$$

L'algorithme CASOM maximise une vraisemblance assimilable à un critère sur une distance moyenne :

[2] On rappelle que la divergence de Kullback-Leibler est toujours positive et s'annule lorsque les deux distributions comparées sont identiques ; mais KL n'est pas une distance symétrique et on a $KL(P||Q) = \sum_j p_j ln(p_j/q_j)$.

Propriété 2
Connaissant la formule liant la logvraisemblance et la divergence, on a :

$$max_\theta \ \mathcal{L}_\mathcal{M}(\theta, \mathbf{C}|\mathcal{D}) \Leftrightarrow min_\theta E_{KL}(\theta, \mathbf{C}|\mathcal{D})$$

□

On peut remplacer la divergence par la distance du χ^2 (d_{χ^2}), on obtient un autre critère très proche de celui étudié ; d_{χ^2} (notée $||\bullet - \bullet||_{\chi^2}$) :

$$E_{\chi^2}(\theta, \mu|\mathcal{D}) = \sum_i f_i \ \sum_k \ \mu_{ik} \ \sum_j \frac{1}{f_j}(f_{j|i} - P_{j|k})^2 = \sum_i f_i \ \sum_k \ \mu_{ik} \ ||f_{i\bullet} - P_{k\bullet}||^2_{\chi^2}$$

En approximant la divergence au deuxième ordre, nous établissons un lien entre E_{χ^2} et E_{Kl}.

Propriété 3 *En notant $\hat{c}_{ik}$ le coefficient d'affectation finale à la convergence de CASOM :*

$$E_{KL}(\hat{\theta}, \hat{\mathbf{C}}|\mathcal{D}) \approx \frac{1}{2} \sum_i \ f_i \ \sum_k \ \hat{c}_{ik} \ \{\sum_j \frac{1}{\hat{P}_{j|k}}(f_{j|i} - \hat{P}_{j|k})^2\}$$

□

Après avoir négligé le terme en $\mathcal{O}(1/N\bullet\bullet)$ dans E_{KL}, ce résultat dérive de l'équivalence asymptotique des statistiques [66]du G^2 et du χ^2, que l'on obtient en écrivant le critère E_{KL} en fonction du log des $\frac{f_{j|i} - \hat{P}_{j|k}}{\hat{P}_{j|k}}$ considérés ici assez proches de zéro pour permettre un développement limité. Mis à part le coefficient multiplicatif $1/2$, nous obtenons que la projection par multinomiales contraintes en carte topologique minimise approximativement[3] en définitive un critère $\hat{E}_{KL}$ de type K-means, doté d'une métrique proche de d_{χ^2}, locale à chacune des classes puisque les P_j sont remplacés par les $\hat{P}_{j|k}$.

Propriété 4
En se restreignant à chacune des classes et en négligeant le lissage bayésien, la distance dans CASOM est localement une distance du χ^2. □

On note $||\bullet - \bullet||_{\chi^2_k}$ la distance du χ^2 adaptative, locale à C_k. Il s'agit plus exactement du χ^2 entre distribution. On remarque que cette distance conserve approximativement la propriété d'équivalence distributionnelle dans les cellules de Voronoï. Le facteur de

[3]Le comportement empirique similaire en classification automatique pour des critères comparables (modalités des variables qualitatives codées en indicatrices) aux précédent a été constaté depuis quelques années en pratique (voir [67]). On en déduit que $\hat{E}_{KL}$ et E_{KL} doivent permettrent de construire des cartes très ressemblantes en pratique. On observe d'ailleurs sur des données réelles E_{χ^2} décroître et atteindre un palier minimum alors que la logvraisemblance du modèle augmente durant l'apprentissage.

lissage induisant des valeurs non nulles, empêche la propriété d'être exacte. En effet, si l'on se restreint à une cellule de Voronoï, et aux données contenues dans cette cellule, on peut considérer les $\hat{P_{j|k}}$ comme les marges du tableau de contingence constituées par les x_i dans la cellule : s'il n'y avait pas lissage par l'ajout d'un facteur additionnel 1, on se retrouverait exactement avec la distance du χ^2, mais seulement pour l'ensemble de données réduit, dans C_k.

Finalement le critère obtenu est le suivant :

$$\tilde{E}_{KL}(\hat{\theta}, \hat{\mathbf{C}}|\mathcal{D}) = \sum_i f_i \sum_k \hat{c}_{ik} \; ||f_{i\bullet} - \hat{P}_{k\bullet}||^2_{\chi^2_k}$$

3.1.4 CASOM : Généralisation non linéaire de l'AFC

Nous avons finalement construit une généralisation non linéaire de l'Analyse de Correspondances. En effet, la modélisation proposée non seulement étend le TPEM aux variables qualitatives en conservant ses relations avec le SOM puisque seule la métrique est modifiée, mais encore, la métrique est approximativement d_{χ^2}, avec adaptation aux classes particulières de documents. On la nomme CASOM pour *SOM by CA* ou AFC par Carte de Kohonen.

Propriété 5
Au sens de la généralisation[4] *définie au chapitre précédent, l'algorithme CASOM est une extension non-linéaire de l'Analyse Factorielle des Correspondances. En effet, il s'agit d'une variante du TPEM pour tableau de contingence, donc de l'algorithme Batch du SOM. La métrique en jeu est la divergence de Kullback-Leibler, assimilable à une distance du χ^2 localement à chaque cellule de Voronoï.* □

3.1.5 Propriétés statistiques de CASOM

L'algorithme proposé possède certaines propriétés en raison de l'hypothèse de distribution multinomiale dans les classes. Nous montrons le caractère gaussien du critère pour des textes tous très longs, et proposons un autre critère appelé divergence-intra étendue quantifiant la qualité de la cartographie par CASOM.

3.1.5.1 Le critère $E_{KL}(\hat{\theta}, \hat{\mathbf{C}}|\mathcal{D})$ pour $min_i N_i$ infini

Nous utilisons dans la suite une propriété connue de la loi multinomiale ; en notant θ^v, le vrai paramètre, $\hat{\theta}$ étant une estimation (par maximum de vraisemblance) :

[4] En remarque, l'Analyse Factorielle des Correspondances est non-linéaire contrairement à l'Analyse en Composantes Principales, donc la propriété précédente n'a pas exactement le même sens que son énoncé pour les cartes de Kohonen. Schématiquement, CASOM est un algorithme qui cherche à construire une surface dans l'espace des distributions empiriques afin d'expliquer d'une certaine manière l'écart à l'indépendance, alors que l'AFC va expliquer cet écart sur des plans factoriels multiples.

Théorème 3
On suppose $\forall j,\ P^v_{j|k} \neq 0$. *On note* θ^v, *le vrai paramètre, si bien que :*

$$\begin{array}{lll} \textit{Si } x_i \sim \mathcal{M}(N_{i\bullet}, \{P^v_{k\bullet}\}) & \textit{, alors } N_{i\bullet} \sum_j (f_{j|i} - P^v_{j|k})^2 / P^v_{j|k} & \stackrel{\mathcal{L}}{\rightarrow} \chi^2_{J-1} \\ \textit{Sinon } (x_i \sim \mathcal{M}(N_{i\bullet}, \{P^{v'}_{k\bullet}\}) \textit{ et } v' \neq v) & \textit{, alors } N_{i\bullet} \sum_j (f_{j|i} - P^v_{j|k})^2 / P^v_{j|k} & \rightarrow \infty \end{array}$$

Ainsi, en assimilant les centres estimés comme les valeurs optimales, ce théorème permet de vérifier une adéquation locale à chaque centre de classe en comparant un vecteur document à son centre attitré. Il serait intéressant de chercher un résultat plus global pour le critère[5] $\tilde{E}_{KL}(\hat{\theta}, \hat{\mathbf{C}}|\mathcal{D})$.

3.1.5.2 Un critère de divergence étendu par analogie avec le SOM

Nous posons un indicateur approché d'auto-organisation adapté au cas particulier des multinomiales. Pour ce faire, nous remplaçons simplement dans $E_{\mathcal{SOM}}$, la distance euclidienne par la distance de Kullback, et les 1/I par les probabilités empiriques f_i :

$$E_{\mathcal{CASOM}}(\theta, \mu|\mathcal{D}) \quad = \sum_i f_i \sum_k \mu_{ik} \sum_l h_{kl} \sum_j f_{j|i} \log \frac{f_{j|i}}{P_{j|l}}$$

En effet,

Propriété 6 *Pour une fonction de voisinage fixe, CASOM minimise approximativement le critère* $E_{\mathcal{CASOM}}(\theta, \mu|\mathcal{D})$, *en ajoutant un lissage.* □

Preuve On a :

$$\begin{array}{ll} & \frac{\partial E_{\mathcal{CASOM}}(\theta,\mu|\mathcal{D}) + \lambda_l(\sum_j P_{j|l} - 1)}{\partial P_{j|l}} = \sum_k \sum_i f_i \mu_{ik} h_{kl} f_{j|i} \frac{1}{P_{j|l}} + \lambda_l = 0 \\ \Rightarrow & P^{n+1}_{j|l} = \frac{\sum_k \sum_i f_i \mu^n_{ik} h_{kl} f_{j|i}}{\sum_k \sum_i \mu^n_{ik} h_{kl} f_i} = \frac{\sum_i h(\xi_l, z^n_i) N_{ij}}{\sum_i h(\xi_l, z^n_i) N_{i\bullet}} \end{array}$$

Par contre, ici, comme pour le SOM, une résolution exacte à la façon des nuées dynamiques serait d'affecter le document x_i à la classe correspondant au minimum sur $f_i \sum_l h_{kl} \sum_j f_{j|i} \log \frac{f_{j|i}}{P_{j|l}}$. Cette affectation est bien différente du maximum a posteriori ; cependant, on constate que pour des centres bien ordonnés, et en cas

[5] Ainsi, on peut imaginer le résultat suivant qu'il faudrait démontrer rigoureusement ; il a peu d'intérêt en réalité dans le cas des données étudiées en pratique puisque les vecteurs textuels sont de grandes dimensions et très peu denses, donc les hypothèses ne sont pas vérifiées.

Pour $min_i N_i$ assez grand, le lissage des multinomiales est négligeable. Sous l'hypothèse H_0 : "Chaque vecteur-document suit la distribution multinomiale de la classe auquel il est affecté", on a pour un grand nombre de documents, tous de grandes tailles (en considérant les probabilités toutes non nulles) :

$$\frac{N_{\bullet\bullet}}{I} \tilde{E}_{KL}(\hat{\theta}, \hat{\mathbf{C}}|\mathcal{D}) \approx J - 1$$

La valeur asymptotique doit diminuer dans le cas de vecteurs creux.

de solution monomodale, les deux types d'affectations sont voisins puisque alors les distances du plus proche voisin aux classes voisines seront peu différentes, amenant à des affectations quasiment identiques. □

L'étude asymptotique du critère approché va permettre d'obtenir un nouveau critère, rappelant un critère proposé pour le SOM. Pour des N_i grands, on obtient en effet, en posant $\hat{z}_i$ l'étiquette de classe définie par CASOM pour x_i :

Propriété 7 *Si on note :*

- $\hat{m}_{\mathcal{CASOM}} = \sum_k (\sum_i \hat{\mu}_{ik} f_i) \sum_{l \neq k} h_{kl} KL(\hat{P}_{k\bullet} || \hat{P}_{l\bullet}) \approx \sum_k \hat{\pi}_k \sum_{l \neq k} h_{kl} KL(\hat{P}_{k\bullet} || \hat{P}_{l\bullet})$
- $\hat{a}_{jk} = h_{k\bullet}(1 + \log \hat{P}_{j|k}) - \log \prod_l \hat{P}_{j|l}^{h_{kl}} \approx \sum_l h_{kl} \log(\hat{P}_{j|k} / \hat{P}_{j|l})$
- $\hat{\sigma}_{\mathcal{CASOM}} = \sqrt{\sum_i f_i [\sum_j \hat{P}_{j|\hat{z}_i} \hat{a}^2_{j\hat{z}_i} - \sum_{j_1} \sum_{j_2} \hat{P}_{j_1|\hat{z}_i} \hat{P}_{j_2|\hat{z}_i} \hat{a}_{j_1\hat{z}_i} \hat{a}_{j_2\hat{z}_i}]}$

Alors, pour $min_i N_{i\bullet}$ assez grand, et sous l'hypothèse que $\hat{\theta}$ est le vrai paramètre, on a le résultat de convergence asymptotique en loi :

$$\frac{E_{\mathcal{CASOM}}(\hat{\theta}, \hat{\mu} | \mathcal{D}) - \hat{m}_{\mathcal{CASOM}}}{\hat{\sigma}_{\mathcal{CASOM}}} \sim \mathcal{N}_1(0, 1)$$

□

Nous avons donné une démonstration rapide de cette propriété en annexe. Une étude plus détaillée reste à faire ; notre cadre applicatif a lieu dans un espace de grande dimension, et ne vérifie généralement pas les hypothèses.

Ce théorème montre le comportement de l'indicateur de bonne répartition sur la grille des documents, ainsi que celui de la classification sous-jacente : une bonne répartition donnera une valeur faible pour cet indicateur, valeur finalement proche de sa valeur asymptotique. Etant donnée la moyenne obtenue, il s'agit bien d'un complément au critère intra. En effet, on retrouve le critère SEL proposé par divers auteurs[61] pour des cartes auto-organisatrices de variables continues, et de centres de classe m_k. On approche de façon informelle les sommes de f_i dans chaque classe à l'aide des coefficients mélangeants. On en déduit la définition suivante :

Définition 27
Un critère de mesure de la qualité de l'auto-organisation pour CASOM s'écrit :

$$\sum_k \hat{\pi}_k \sum_{l \neq k} h(\xi_k, \xi_l) KL(\hat{P}_{k\bullet} || \hat{P}_{l\bullet})$$

Ce critère est assez semblable à celui pour le SOM, puisqu'il suffit de remplacer la distance euclidienne par la divergence, et d'assimiler les centres de classes m_k aux

$P_{k\bullet}$. Il s'ajoute par contre une pondération prenant en compte la probabilité a priori des centres, non existant dans le critère original. Ce résultat permet d'obtenir un intervalle de confiance du critère et de mieux comparer deux valeurs différentes. C'est une statistique de test asymptotique. Cependant, l'hypothèse de limite asymptotique généralement pas atteinte, nous amène à proposer en alternative une méthode empirique basée sur une simulation de type Monte-Carlo, qui est présentée dans la section suivante, ainsi que divers autres indicateurs statistiques.

3.1.6 Résumé sur la méthode CASOM

Nous avons montré sur un cas particulier qu'une modélisation probabiliste est équivalente à l'usage d'une distance, une distance du χ^2 locale, que nous avons mis en évidence. La divergence est une bonne alternative à la distance euclidienne. Pondérée par la taille des textes, elle permet de prendre en compte des textes de longueur différente, et rappelle le χ^2 pondéré décomposé par l'AFC. La prise en compte du nombre de mots dans les textes est souvent inexistante dans les algorithmes courants ; c'est donc un point important de la projection à base de multinomiales. Nous avons également explicité une limite asymptotique du critère. Ce résultat est important lorsque l'on travaille en grande dimension, comme en analyse textuelle. Il permet de prévoir le comportement pratique des critères proposés. Ici, pour un grand nombre de documents, tous très grands on constate que l'on maîtrise la comparaison de valeurs du critère : la loi dégagée donne directement une estimation de la pertinence entre des écarts. Il reste à les étudier plus en détail dans nos cas plus particuliers (lissage bayésien et I, $N_{i\bullet}$ petits). En effet, les données ne vérifient pas toujours les hypothèses asymptotiques. Il serait intéressant d'essayer d'établir des résultats en petit échantillon, en tenant compte du biais bayésien. On pourrait également chercher le comportement de la distance en fonction de l'écart à la vraie loi, propriété importante puisque l'on dispose rarement des estimations exactes. Un autre intérêt de la méthode présentée est certainement de construire un pont entre les méthodes de carte auto-organisatrice et les méthodes d'AFC, à l'aide de la loi multinomiale. Cette idée semble un bon point de départ à l'élaboration d'algorithmes généralisant l'AFC. Un problème commun au SOM, TPEM et CASOM, est la difficulté de réglage des paramètres. Il existe des moyens relativement lourds pour y remédier. C'est pourquoi, nous proposons dans la suite des façons de formaliser explicitement le voisinage. Auparavant, nous expliquons des moyens d'étudier les projections obtenues par CASOM.

3.2 Compléments à l'étude d'une cartographie par CASOM

Nous allons définir différents critères qui permettent de quantifier d'un point vu statistique ou de celui de la théorie de l'information, la dépendance entre les multinomiales. D'abord, nous proposons une méthode visuelle en construisant une méthode d'analyse factorielle dédiée à la visualisation des multinomiales, son lien avec l'analyse

factorielle est alors présenté. A cette occasion, nous posons un tableau de contingence associé à la partition obtenue par CASOM, dérivé de la méthode d'estimation par maximum de vraisemblance du mélange de multinomiales avant la normalisation pour obtenir les centres multinomiaux. Ce nouveau tableau explicité fournit les bases d'un classique test du χ^2 afin d'étudier la dépendance des facteurs multinomiaux de ce point de vue, et introduire un critère d'information, la somme de l'entropie des facteurs. Enfin, il est expliqué comment parvenir à un test statistique cohérent avec le critère de la variance-intra étendue aux voisins.

3.2.1 Projections factorielles

Nous proposons de projeter les centres des multinomiales par l'analyse factorielle des correspondances. Il est choisi pour pondération de chacune des classes, la probabilité du facteur multinomial, et pour métrique, l'inverse des probabilités des mots. La projection permet de montrer les dépendances entre facteurs multinomiaux et de juger de la bonne convergence de l'algorithme. Il s'agit d'un complément à la représentation par U-matrice (voir [38]) qui montre[6] la divergence plutôt que la distance du χ^2. Il est alors possible de représenter les mots en éléments supplémentaires.

3.2.1.1 Méthode de projection de centres multinomiaux

Plus formellement, on note P_K, la matrice diagonale de rang K et d'éléments diagonaux $\hat{\pi}_k$. La matrice des profil-lignes étudiée est celle des facteurs multinomiaux, soit la matrice $D_K = [\hat{P}_{j|k}]_{k,j}$ de K lignes et J colonnes. Nous appliquons donc une projection factorielle, avec la métrique du χ^2 $(f_{\bullet j})$, et la matrice des poids P_K, sur la matrice de profil-lignes D_K, réduction de D. Les vecteurs propres solutions diagonalisent la matrice :

$$D_K^T P_K^{+1} D_K P_J^{-1}$$

On remarque que cette projection factorielle n'est pas tout à fait une AFC car les pondérations proposées ne sont pas les marges normalisées du tableau de contingence D_K. On peut montrer cependant une équivalence sous certaines hypothèses. Dans la suite, nous définissons un tableau de contingence particulier.

3.2.1.2 Lien avec une AFC

En effet, la projection ne décompose pas l'inertie $I_{\mathcal{M}}$ (au facteur additionnel 1 près) proportionnellement à un χ^2 contrairement à l'AFC. Cela provient principalement du fait que les multinomiales ne se mélangent pas comme la somme normalisée du vocabulaire présent dans les textes, mais plutôt par normalisation des nombres de documents

[6] Pour notre méthode particulière, nous remplaçons la distance euclidienne utilisée pour la U-matrice du SOM avec distance euclidienne par la divergence symétrisée qui a bien un sens ici. On use d'une symétrisation pour corriger le fait que KL n'a justement pas la propriété de symétrie.

présents dans leur classe correspondante. En effet, on remarque dans l'algorithme d'estimation des facteurs multinomiaux que les $\hat{P}_{j|k}$ sont calculés par normalisation d'un tableau de contingence à chaque pas de l'algorithme :

$\sum_i \hat{\mu}_{i1} N_{ij} + 1$	$\cdots$	$\cdots$	$\cdots$	$\sum_i \hat{\mu}_{i1} N_{iJ} + 1$	$\sum_i \hat{\mu}_{i1} N_{i\bullet} + J$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\sum_i \hat{\mu}_{ik} N_{ij} + 1$	$\cdots$	$\cdots$	$\sum_i \hat{\mu}_{ik} N_{i\bullet} + J$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\sum_i \hat{\mu}_{iK} N_{ij} + 1$	$\cdots$	$\cdots$	$\cdots$	$\sum_i \hat{\mu}_{iK} N_{iJ} + 1$	$\sum_i \hat{\mu}_{iK} N_{i\bullet} + J$
$\sum_k \sum_i \hat{\mu}_{ik} N_{ij} + K$	$\cdots$	$\cdots$	$\cdots$	$\sum_k \sum_i \hat{\mu}_{ik} N_{iJ} + K$	$\sum_k \sum_i \hat{\mu}_{ik} N_{i\bullet} + KJ$

C'est l'AFC de ce tableau qu'il faudrait calculer pour étudier les liaisons exactes au sens du χ^2 entre les facteurs multinomiaux. Si l'on note la matrice diagonale $\tilde{P}_K^{-1}$ d'éléments $\frac{\sum_i \hat{\mu}_{ik} N_{i\bullet} + J}{\sum_i \hat{\mu}_{i\bullet} N_{i\bullet} + KJ}$ ainsi que $\tilde{P}_J^{-1}$, la matrice d'élément $\frac{\sum_i \hat{\mu}_{i\bullet} N_{ij} + K}{\sum_i \hat{\mu}_{i\bullet} N_{i\bullet} + KJ}$, une application directe de l'AFC correspond à diagonaliser la matrice :

$$D_K^T \tilde{P}_K^{+1} D_K \tilde{P}_J^{-1}$$

Notre approche préfère prendre en compte les vrais poids qui sont affectés ($\hat{\pi_k}$) aux facteurs multinomiaux, et suppose disposer uniquement de leur estimation (tableau inconnu). Dans un cas asymptotique et à vocabulaire équiréparti (N_i tous égaux), il est clair que la méthode décompose approximativement le χ^2 d'indépendance puisque les marges s'identifient aux pondérations proposées. Par conséquent, on ne conserve pas la première valeur propre tout comme en AFC, puisque le vecteur propre correspondant est ici approximativement le vecteur moyen d'élément P_j. On peut considérer la première valeur propre comme une mesure de l'écart (sous l'hypothèse d'une estimation parfaite, puisqu'en générale, cette valeur propre sert normalement à juger de la précision numérique obtenue en AFC) entre la projection proposée et celle issue directement de l'AFC.

3.2.1.3 Remarques sur la projection

La représentation proposée est également à rapprocher de la méthodologie d'Elemento et LeChevallier dans [68] qui préconisent une projection factorielle. Ils étudient une projection de la carte dont les centres sont vus comme individus supplémentaires, à partir d'une ACP sur l'ensemble des individus. Ils proposent en outre une initialisation à partir de ces plans pour éviter l'utilisation d'une fonction de voisinage à large variance dès le début de l'algorithme; en effet, une mauvaise initialisation et un apprentissage mal adapté peuvent conduire à une carte dont la topologie générale respecte peu la distribution des données originales, l'auteur construit une initialisation pré-organisée grâce à l'ACP. Dans notre cas, il est plutôt proposé d'observer après entraînement les dépendances au sens du χ^2 des centres des classes. Cela permet d'observer par la même occasion les variables mais aussi les individus projetés en

éléments supplémentaires.

Cependant, si ce genre[7] de visualisation paraît pertinent pour des tailles de cartes réduites, elle le devient moins pour de grandes cartes, dont la complexité devrait augmenter, le premier plan factoriel devenant par conséquent bien moins interprétable. Des indicateurs chiffrés apparaissent nécessaires, d'autant plus que ce genre d'indicateur doit permettre idéalement de vérifier la bonne projection des données. Dans les parties suivantes, il est proposé un ensemble d'indicateurs naturels non factoriels.

3.2.2 Critères de dépendance

Il apparaît intéressant de tracer en complément des indicateurs de vraisemblance, un critère statistique de mesure de dépendance : le χ^2 sur le tableau de contingence sous-jacent aux multinomiales. Ce critère doit a priori décroître au fur et à mesure de l'estimation des multinomiales contraintes puisque l'on force une dépendance spatiale qui diminue à chaque pas d'itération.

On rappelle la définition du test d'indépendance pour tableau de contingence, plus généralement appelé test du χ^2 qui permet de vérifier l'adéquation d'une distribution empirique à une loi théorique. Ici, on vérifie si les deux variables qualitatives croisées dans le tableau sont indépendantes ou non : en cas d'indépendance la probabilité d'occurrence d'une cellule ne dépend que des marginales pour les ligne et colonne correspondantes.

Définition 28 ([66, 15])
*Le **test du** χ^2 pour un tableau de contingence de cellules de comptage n_{kj} teste l'hypothèse H_0 : "Les lignes ont même distribution" contre H_1 : "Les lignes sont de distribution différentes ". Pour ce faire, on compare les effectifs constatés n_{kj} avec les effectifs attendus ou effectifs théoriques (estimés ici par $n_{k\bullet}n_{\bullet j}n_{\bullet\bullet}^{-1}$) $n_{k\bullet}p_j$. Sous hypothèse H_0, on a la statistique :*

$$D^2 = \sum_k \sum_j \frac{(n_{kj} - n_{k\bullet}p_j)^2}{n_{k\bullet}p_j} \sim \chi^2_{(K-1)(J-1)}$$

Une valeur trop grande de D^2 conduit à rejeter l'hypothèse H_0, et donc de conclure à la non-indépendance des lignes.

[7] On peut remarquer que la projection n'est rien d'autre qu'une analyse discriminante pour variable qualitative. En effet, en analyse discriminante pour des vecteurs de variables continues, la projection maximisant la variance inter projetée est équivalente à l'ACP des centres pondérés par le nombre d'individus présent dans chaque classe. Or, l'estimation du mélange classifiant donne des estimations des poids des classes et valeurs des distributions comparables. Cette représentation enfin est rapide à calculer puisqu'il s'agit de réaliser une AFC approchée sur un nombre réduit d'individus, les facteurs multinomiaux. Les cartes factorielles construites doivent permettre d'affiner une analyse en segmentant le treillis de neurones en thématiques homogènes.

Nous utilisons l'approche de ce test pour définir les nouveaux indicateurs de mesure de liaison qui suivent. Il s'agit de calculer le χ^2 du tableau des vecteurs multinomiaux. Ainsi, nous mesurons combien ces vecteurs ont des lois comparables ou non, et d'un point de vue global.

Définition 29 *Un critère de dépendance des multinomiales contraintes s'écrit comme le χ^2 du tableau sous-jacent :*

$$D^2(\theta|\mathcal{D}) = \sum_{k:\sum_i \mu_{ik}>0} \sum_j \frac{\left(\sum_i \mu_{ik} N_{ij} + 1 - \frac{(\sum_i \mu_{ik} N_{i\bullet} + J) \times (\sum_i \mu_{i\bullet} N_{ij} + K)}{\sum_i \mu_{i\bullet} N_{i\bullet} + KJ} \right)^2}{\frac{(\sum_i \mu_{ik} N_{i\bullet} + J) \times (\sum_i \mu_{i\bullet} N_{ij} + K)}{\sum_i \mu_{i\bullet} N_{i\bullet} + KJ}}$$

Un autre critère sensible à l'organisation de la grille est le critère d'entropie. En effet, le lissage induisant l'auto-organisation crée une dépendance spatiale par construction. Aussi il doit certainement augmenter l'entropie. Cela se justifie de façon informelle et intuitive par une diminution du désordre au niveau de la répartition des probabilités, et justement une propension des vecteurs multinomiaux à se réduire à la moyenne du nuage. L'hypothèse de recuit sous-jacent a déjà été faite pour le SOM, mais ici, nous avons une expression explicite étant donnée la nature[8] des données traitées :

Définition 30 *Un critère de mesure de l'organisation des multinomiales est la somme des entropies des facteurs multinomiaux :*

$$\mathcal{H}(\theta|\mathcal{D}) = \sum_{k:\sum_i \mu_{ik}>0} \sum_j \frac{\sum_i \mu_{ik} N_{ij} + 1}{\sum_k \sum_i \mu_{ik} N_{i\bullet} + K} \log \left(\frac{\sum_i \mu_{ik} N_{ij} + 1}{\sum_k \sum_i \mu_{ik} N_{i\bullet} + K} \right)$$

Cela explique de façon intuitive pourquoi la surface discrète formée des centres de classe disposés dans l'espace des données a tendance à être de faible surface au début de l'estimation pour devenir de plus en plus grande jusqu'à épouser étroitement les données. Ce critère de désordre est d'une certaine manière un critère de liaison spatiale étant donnée la présence de la somme sur les classes. En effet, de façon informelle, plus les centres de classes sont différents, moins ils sont comparables à la valeur moyenne du nuage (vecteur de composantes P_j), et donc, plus l'entropie est faible. Enfin, il serait également possible de mesurer la dépendance d'un point de vue informationnel à partir de l'information mutuelle. Or, cette dernière n'est rien d'autre que le χ^2 de vraisemblance (à un facteur proportionne et additif près) ou G^2, de même loi asymptotique que D^2.

A partir de ces définitions, nous posons finalement les deux conjectures suivantes :

[8] En remarque, il est tout à fait envisageable d'exprimer ces critères dans le cas gaussien (cf. TPEM) en usant notamment de l'entropie différentielle et de corrélations linéaires.

CONJECTURES

Premièrement, nous conjecturons que l'entropie d'un pas CASOM est inférieure à celle d'un pas CEM. Dans la littérature, il est couramment fait l'hypothèse que l'algorithme du SOM introduit un niveau d'entropie décroissant à l'aide de la fonction de voisinage, CASOM permet de l'écrire ainsi :

- $\mathcal{H}_{\mathcal{CASOM}}(\theta|\mathcal{D}) \geq \mathcal{H}_{\mathcal{CEM}}(\theta|\mathcal{D})$

Deuxièmement, l'hypothèse sur la dépendance statistique est par contre une nouvelle manière de considérer le SOM. En effet, étant donné la propriété de généralisation de l'AFC par CASOM, nous sommes en mesure d'exprimer directement la dépendance statistique des multinomiales par le biais du tableau de contingence. Le voisinage introduit lors de l'estimation des centres multinomiaux entraîne en effet des relations linéaires entre voisins : des dépendances statistiques doivent apparaître :

- $D^2_{\mathcal{CASOM}}(\theta|\mathcal{D}) \geq D^2_{\mathcal{CEM}}(\theta|\mathcal{D})$

Ces deux inégalités pourraient directement porter sur deux pas consécutifs de CASOM, hypothèse encore plus forte. Pourtant, ces critères ne prennent pas en compte explicitement les particularités de la contrainte locale introduite lors de l'estimation des multinomiales. C'est pourquoi, nous avons construit tout spécialement un nouveau test statistique dans la partie précédente. Dans la suite, nous proposons des tests statistiques exempts des hypothèses asymptotiques sur les vecteurs de comptage.

3.2.3 Vers un test statistique pour une évaluation de la qualité spatiale ou une initialisation automatique, à l'aide d'une simulation Monte-Carlo

Des auteurs[69] ont présenté récemment une méthode de *boostrapping* pour évaluer la qualité de projection d'une carte auto-organisatrice. Pour cela, ils réalisent un rééchantillonnage des données, et recalculent le critère de variance-intra, ainsi qu'un critère de voisinage, la moyenne empirique évaluant le nombre moyen de fois que deux individus sont voisins. Alors, ils procèdent à un test binomial qui vérifient si deux individus ne sont pas voisins par hasard, étant donné leur chance de l'être, comme le rapport de la surface du voisinage considéré avec la surface totale de la carte. Ce genre de méthode reconnue pour son efficacité et sa fiabilité n'en demeure pas moins très coûteuse en temps machine, puisque l'échantillonnage doit se faire sur l'ensemble des données. Nous proposons une méthode moins intensive qui ne nécessite que des calculs sur la carte finale.

La question que nous nous posons ici est d'évaluer la bonne répartition des classes plutôt que celle des individus. C'est-à-dire, étant donnée la projection obtenue, les classes sont-elles bien placées sur la carte ? Pour y répondre nous procédons également à une approche Monte-Carlo. C'est-à-dire que l'on suppose que la répartition obtenue est l'observation d'un phénomène aléatoire, et nous allons chercher à estimer son comportement. Nous modifions au hasard la répartition des classes un grand nombre de fois, et évaluons à chaque fois un critère de répartition spatiale entre classes :

$$C_{\mathcal{M}}(\{P_{k\bullet}\}_k) = \sum_k \pi_k \sum_l h_{kl}\, KL(P_{k\bullet}||P_{l\bullet})$$

En effet, calculer de façon exhaustive l'ensemble de toutes les configurations possibles, c'est-à-dire K !, est impossible pour les nombres élevés de classes rencontrés en pratique. L'histogramme des valeurs permet de calculer la probabilité (empirique) de meilleure répartition. A un seuil donné, on est capable de décider[9] si la projection correspond probablement à une bonne répartition des classes ou non. Il s'agit de l'algorithme suivant :

Algorithme 10

Test Monte-Carlo de qualité

Init	*Variables* $P_{k\bullet}^{MC}$ $k \in \{1, 2, \cdots, K\}$,
Pour n de 1 à T	*Début Répéter* —–
Affectation	$P_{k\bullet}^{MC} \leftarrow \hat{P}_{k\bullet}$
Config. aléatoire	*Répartition aléatoire des* $P_{k\bullet}^{MC}$ *en* $P_{k\bullet}^{MC,n}$
Evaluation	*Calcul de* $C_{\mathcal{M}}$ *sur les* $P_{k\bullet}^{MC,n} \Rightarrow C_{\mathcal{M}}^n$,
Jusque T	*Fin répéter* —–
Proba empirique	*Evaluer l'histogramme* $Hist(C_{\mathcal{M}}^{\bullet})$ *des* $C_{\mathcal{M}}^n$,
	Calcul de la probabilité empirique de $C_{\mathcal{M}}(\{\hat{P}_{k\bullet}\}_k)$ *pour* $Hist(C_{\mathcal{M}}^{\bullet})$

Cette approche n'est pas sans rappeler certaines méthodes de statistique spatiale employées en géostatistique pour évaluer le degré de cohérence de la répartition d'une grandeur sur une zone géographique. On parle de Test de permutation pour mesurer l'auto-corrélation spatiale à partir de la statistique de produit croisé. Le test de Geary et l'indice de Moran sont envisageables ici, afin de vérifier la cohérence spatiale. Il est clair que la procédure proposée permet de déterminer si la qualité de la carte est satisfaisante, et d'autant mieux si la densité est complexe car dans ce cas, la distorsion globale de la projection risque d'être importante. Par contre, elle risque de détecter difficilement des petits défauts qui demandent peu de modification de la carte puisque

[9] On calcule la "*P-value*" empirique correspondant à E_{CASOM}, i.e. la proportion de cartes permutées t.q. $C_{\mathcal{M}} > E_{CASOM}$.

dans ce cas, la méthode est limitée par le nombre T d'essais effectués. De plus, on en déduit directement un algorithme d'ordonnancement des centres de classes sur une carte : il suffit d'employer l'algorithme précédant en plaçant aléatoirement les centres sur une carte puis en choisissant la meilleure carte obtenue. Un lissage à l'aide des voisins proches permet d'obtenir ensuite une carte qui pourrait faire office d'initialisation notamment pour CASOM.

3.2.4 Distribution discrète bivariée : vers une représentation sémantique du vocabulaire

Les facteurs multinomiaux contiennent les probabilités des mots du vocabulaire. Etant donné que ces probabilités sont bornées et contiennent une information spatiale du fait de la carte auto-organisatrice, nous proposons l'utilisation ad'hoc de cette propriété en visualisant les probabilités $\hat{P}_{j|k}$ à j fixé, pour k allant de 1 à K, sur une carte. Comme cela on peut détecter des zones d'utilisation fréquente. Cela a une application directe d'extraction de connaissance, et de recherche d'information. En effet, en sommant de telles cartes pour l'ensemble des mots employés dans une question x_q, on peut en déduire les zones de fortes activations de la carte. Contrairement au SOM, nous obtenons un résultat directement interprétable. En effet, nous avons une représentation qui se rapproche d'une distribution bivariée discrète, alors que pour le SOM original, les valeurs ne sont pas bornées, mais dans $\mathcal{R}$. On a une carte $K = LX \times LY$:

$\hat{P}_{j\|1}$	$\cdots$	$\cdots$	$\cdots$	$\hat{P}_{j\|LX}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\hat{P}_{j\|k}$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\hat{P}_{j\|K-LX}$	$\cdots$	$\cdots$	$\cdots$	$\hat{P}_{j\|LX\times LY}$

On remarque pourtant qu'il ne s'agit pas d'une distribution de probabilité, pour la raison simple que $\sum_k \hat{P}_{j|k} \neq 1$. En effet, les facteurs multinomiaux ne somment que pour J. Rien n'empêche d'y remédier en calculant au lieu de $\hat{P}_{j|k}$, une normalisation par rapport aux classes du tableau de contingence sous-jacent aux multinomiales, et obtenir les cellules :

$$\frac{\sum_i \hat{\mu}_{ik} N_{ij} + 1}{N_{\bullet j} + K}$$

Une telle normalisation fournit une solution plus interprétable puisque l'on observe alors la carte obtenue comme une véritable distribution discrète bivariée. On peut mesurer l'entropie d'un mot donné dans les classes de documents : au niveau de la représentation offerte à une visualisation éventuelle. On peut parler d'un certain niveau sémantique en assimilant les modes comme autant de thématiques particulières où s'emploie un mot donné dans tel ou tel cas de figure. On peut définir également une mesure de distance entre deux cartes en comparant leur distribution. En définitive,

cette représentation doit montrer le degré de pertinence d'un mot pour qualifier une thématique : un mot peu spécialisé apparaîtra dans de nombreuses classes contrairement à un mot caractéristique de thématiques.

Définition 31 *Un critère de mesure de la qualité de caractérisation thématique d'un mot est l'entropie de la distribution. On l'appelle* ***entropie cartographique de*** v_j *:*

$$\mathcal{H}(v_j|\hat{\theta}) = \sum_k \frac{\sum_i \hat{\mu}_{ik} N_{ij} + 1}{N_{\bullet j} + K} \log\left(\frac{\sum_i \hat{\mu}_{ik} N_{ij} + 1}{N_{\bullet j} + K}\right)$$

De même,

Définition 32 *Un critère de mesure entre deux mots est la distance entre distribution. On l'appelle* ***degré de synonymie sémantique entre*** v_{j_1} ***et*** v_{j_2} *:*

$$D(v_{j_1}, v_{j_2}|\hat{\theta}) = \sum_k \frac{\left(\frac{\sum_i \hat{\mu}_{ik} N_{ij_1}+1}{N_{\bullet j_1}+K} - \frac{\sum_i \hat{\mu}_{ik} N_{ij_2}+1}{N_{\bullet j_2}+K}\right)^2}{\left(\frac{\sum_i \hat{\mu}_{ik} N_{ij_1}+1}{N_{\bullet j_1}+K} + \frac{\sum_i \hat{\mu}_{ik} N_{ij_2}+1}{N_{\bullet j_2}+K}\right)}$$

Il est clair qu'au lieu de calculer une distance entre deux cartes de distribution a posteriori, il est possible de vérifier visuellement la superposition des modes entre deux cartes, et de conclure qualitativement. Nous présenterons de telles cartes dans la partie suivante qui illustrent CASOM sur des données textuelles. La méthode de cartographie présentée aboutit finalement à une certaine mesure des liaisons statistiques entre les différents mots du vocabulaire. **On aboutit également à une représentation simultanée des distributions du vocabulaire et du corpus, ainsi que de leur projection moyenne.**

On peut imaginer des pondérations en fonction de l'importance d'une classe, et également l'emploie de distance variée. De même l'entropie peut se pondérer et être modifiée. Les résultats que nous obtenons en pratique, en affichant directement les probabilités des multinomiales, paraissent qualitativement satisfaisants d'un point de vue cohérence empirique spatiale des zones obtenues. C'est pourquoi nous avons gardé les probabilités originales pour les représentations. En effet, il est possible de comparer à l'oeil nu les représentations de deux mots du vocabulaire, et de vérifier qu'ils sont employés dans les mêmes zones ou non, et identifier ces zones pour un accès aux documents qui y sont contenus. De plus, cette représentation permet de définir de nouveaux indicateurs dont la pertinence reste à voir. Ainsi, on peut imaginer une pseudo-entropie ($-\sum_k \hat{P}_{j|k} log(\hat{P}_{j|k})$) de la carte de m_j, ou bien une pseudo-distance qualifiable de sémantique entre deux mots, en évaluant une distance adéquate (par exemple $\sum_k \frac{(\hat{P}_{j|k}-\hat{P}_{j|k})^2}{\hat{P}_{j|k}+\hat{P}_{j|k}}$) entre les représentations. L'étude de tels critères devrait permettre de comprendre l'information contenue dans cette représentation, notamment vis-à-vis de la fréquence de m_j. Il est clair que la peudo-entropie est corrélée avec la fréquence totale, résultat peu informatif au demeurant.

3.2.5 Conclusion sur les critères spatiaux

Nous avons introduit de nouveaux indicateurs statistiques en adaptant des critères connus pour le SOM original, ou en construisant de nouveaux afin d'essayer d'appréhender le comportement des cartes auto-organisatrices, et surtout d'interpréter non seulement les résultats obtenues mais également le degré de qualité, voire de confiance que l'on peut porter à des telles cartes. L'usage des multinomiales apporte un regard nouveau dans le domaine. Etudier d'un point de vue statistique la construction des cartes, à l'aide de ces indicateurs ad'hoc doit ouvrir des perspectives nouvelles, notamment du point de vue de l'étude de la cohérence spatiale d'une grande carte lorsque les données ne sont pas étiquetées.

3.3 TNEM, algorithme en analogie à la segmentation d'image

La segmentation d'image consiste en la recherche de zones homogènes sur une image fixe. Bien que d'un domaine totalement différent de l'analyse de données, la segmentation présente par certains aspects, notamment la notion de voisinage, des points communs importants. C'est pourquoi, nous établissons un parallèle direct entre les deux domaines afin d'en dégager d'éventuels nouveaux algorithmes de cartes auto-organisatrices. En effet, le domaine de traitement de l'image a donné lieu à un travail intensif, et les nombreux résultats obtenus à cette occasion pourraient être adaptés et étendus à notre problématique. D'autant plus que les questions de dimensions, i.e. taille d'échantillon et de l'espace sont également rencontrées en image. Il s'agit ici de chercher de nouvelles[10] façons de modéliser le voisinage des cartes auto-organisatrices. Ainsi, nous considérons les deux processus de segmentation et de carte auto-organisatrice comme deux méthodes apparentées : la segmentation cherche un label optimal en respectant un voisinage entre individus tandis que la carte auto-organisatrice cherche ce label tout en respectant un voisinage entre classes. Il s'agit finalement de problématiques comparables de ce point de vue mais avec des contraintes différentes.

L'analogie [segmentation d'image] VS [carte auto-organisatrice] se définit comme la recherche d'une partition des individus sous une contrainte de voisinage précise :

- segmentation d'image : voisinage fixé sur les individus,
- carte auto-organisée : voisinage fixé sur les classes.

Nous introduisons la définition suivante :

Définition 33
Une transformation d'un algorithme de segmentation d'image en un algorithme de carte auto-organisatrice est la modification du voisinage des individus en celui de classes.

[10] L'algorithme obtenu dans la suite s'affranchit d'ailleurs de la distribution de classe et peut s'appliquer à des données quelconques, continue ou discrète, ou les deux à la fois.

Ce procédé est appliqué ci-dessous pour le cas des algorithmes gibsiens de segmentation d'image avec estimation approchée par champ moyen. D'autres algorithmes de carte s'obtiennent en choisissant d'autres algorithmes de segmentation, encore faut-il les valider pour s'assurer de leur exactitude. Nous n'avons en effet pas prouvé que tout algorithme de segmentation donne un algorithme de construction de carte : ce travail reste à faire au moins pour des classes de méthodes. Dans la suite, nous rappelons le modèle employé en image puis le modifions et étudions l'algorithme obtenu. Des simulations sont présentées dans le chapitre suivant.

3.3.1 Modèles de champ aléatoire de Markov

3.3.1.1 Introduction aux MRFs

Soit[70] $\mathbf{X}_{\mathcal{I}} = (X_i)_{i\in\mathcal{I}}$, une suite de v.a. pour $\mathcal{I} = [1, 2, \cdots, I]$. On note toute restriction de $\mathbf{X}_{\mathcal{I}}$ par $\mathbf{X}$ indicé[11] par le sous-ensemble restreint de $\mathcal{I}$.

Définition 34 *On dit que $\boldsymbol{X}$ est un champ aléatoire de Markov (MRF) de paramètres ϕ si :*

1. *Les réalisations de $\boldsymbol{X}$ sont non nulles,*
2. *on a : $P(X_i|\boldsymbol{X}_{\mathcal{I}-\{i\}}, \phi) = P(X_i|\boldsymbol{X}_{\mathcal{N}_i}, \phi)\ \forall i \in \mathcal{I}$*

On a noté $\mathcal{N}_i$ un proche voisinage du site i. Il est clair que cette relation de dépendance relative à un voisinage rappelle les propriétés des cartes auto-organisatrices. Ici, on présente ce modèle tel qu'il est utilisé en analyse statistique dans le domaine du traitement de l'image.

Grâce au théorème de Hammersley-Clifford, si $\mathbf{X}$ est un MRF, alors, c'est également un distribution de Gibbs. Une distribution de Gibbs existe sur un graphe donné représentant les dépendances entre v.a. Chaque noeud du graphe s'appelle un site. On note U, une fonction d'énergie définie comme une somme de potentiels U_c, qui sont des fonctions de v.a., restriction de $\mathbf{X}$ à une clique (ensemble de noeuds voisins) donnée c du graphe. Soit, en notant $\mathcal{I}_c$ la restriction de $\mathcal{I}$ à la clique c :

Définition 35 *Une distribution de Gibbs de paramètres ϕ vérifie :*

$$P(\boldsymbol{X}|\phi) = Z(\phi)^{-1} exp(-U(\mathbf{X})|\phi) \text{ où } U(\mathbf{X}) = \sum_c U_c(\boldsymbol{X}_c|\phi), \text{ et } Z = \sum_{\mathbf{x}} exp(-U(\mathbf{x}|\phi))$$

Cette expression explicite des dépendances locales (spatiales) au niveau des cliques. Pourtant l'expression complète du champ de Gibbs demeure lourde à cause de la normalisation qui somme sur un grand nombre de variables en pratique. Les champs de Gibbs sont connus en statistique physique pour notamment modéliser les interactions d'un grand nombre de particules.

[11] L'indice $\mathcal{I}$ est généralement implicite dans la suite.

3.3.1.2 L'approximation des champs moyens

On peut voir qu'il est relativement aisé de calculer la probabilité d'une réalisation sur un site conditionnellement à la réalisation des autres sites en fonction des uniques sites voisins lorsque X_i prend ses valeurs dans $\mathcal{K} = \{1, 2, \cdots, K\}$; le facteur de normalisation disparaît dans le quotient :

$$\begin{aligned} P(x_i|\mathbf{x}_{\mathcal{I}-\{i\}}, \phi) &= \frac{P(x_i, \mathbf{x}_{\mathcal{I}-\{i\}}, \phi)}{\sum_{k\in\mathcal{K}} P(k, \mathbf{x}_{\mathcal{I}-\{i\}}, \phi)} \\ &= \frac{Z(\phi)}{Z(\phi)} \frac{exp(\sum_{c:i\in c} -U_c(x_i, x_{c-i}|\phi)) exp(\sum_{c:i\in\bar{c}} -U_c(x_c|\phi))}{\sum_{k\in\mathcal{K}} exp(\sum_{c:i\in c} -U_c(k, x_{c-i}|\phi)) exp(\sum_{c:i\in\bar{c}} -U_c(c|\phi))} \\ &= \frac{exp(\sum_{c:i\in c} -U_c(x_i, x_{c-i}|\phi))}{\sum_{k\in\mathcal{K}} exp(\sum_{c:i\in c} -U_c(k, x_{c-i}|\phi))} \end{aligned}$$

Même si cette probabilité conditionnelle reste peu lourde à calculer, le passage à la marginale demeure numériquement impossible. Il a été proposé dans la littérature[70, 71, 72, 73] une approximation qui donne d'excellents résultats, l'approximation des champs moyens. Il s'agit de supposer que les voisins d'un site ne sont plus aléatoires, mais déterministes, fixés à leur valeur moyenne $\bar{x_i}$:

$$P(x_i|\phi) \approx P_{cm}(x_i|\phi) = \frac{exp(\sum_{c:i\in c} -U_c(x_i, \bar{x}_{c-i}|\phi))}{\sum_{k\in\mathcal{K}} exp(\sum_{c:i\in c} -U_c(k, \bar{x}_{c-i}|\phi))}$$

Pour conserver une cohérence avec l'hypothèse de champ moyen, on calcule les moyennes sur les x_i, ce qui donne des équations du point fixe :

$$\bar{x_i} = \frac{\sum_{x_i\in\mathcal{K}} x_i \times exp(\sum_{c:i\in c} -U_c(x_i, \bar{x}_{c-i}|\phi))}{\sum_{k\in\mathcal{K}} exp(\sum_{c:i\in c} -U_c(k, \bar{x}_{c-i}|\phi))} \ \forall i$$

Ceci constitue les équations du point fixe. En alternative à cette méthode, il est par exemple possible d'approcher ces valeurs à partir de l'échantillonneur de Gibbs[74] qui permet de simuler des réalisations de X_i à partir des seuls voisins dans la clique de X_i. En bref, il s'agit de générer une suite de valeurs à partir d'une initialisation quelconque ; l'algorithme modifie une composante à la fois du vecteur précédemment généré pour en créer un nouveau dont la composante à modifier est la plus probable conditionnellement aux voisins restés constants. L'algorithme de Metropolis en est une alternative.

3.3.1.3 Modèle de segmentation d'image

On décrit le modèle de segmentation usuellement utilisé en segmentation d'image. Ce modèle pose à la façon des modèles de carte auto-organisatrice une contrainte sur le voisinage des pixels de l'image. Soit les variables aléatoires $X_1, X_2, \cdots, X_I$, v.a. i.i.d. regroupées dans le vecteur $\mathbf{X}$, dont on dispose de la réalisation $x_1, x_2, \cdots, x_I$ ($\mathbf{x}$ correspondant à $\mathcal{D}$ dans la notation employée jusqu'ici). De même les variables de classe se notent $Z_1, Z_2, \cdots, Z_I$ regroupées dans le vecteur $\mathbf{Z}$; elles prennent leur valeur dans $\mathcal{Z} = \{1, 2, \cdots, K\}$. Comme dans un modèle de mélange, on suppose :

$$P(\mathbf{X}|\mathbf{Z}, \theta) = \prod_i P(X_i|Z_i, \theta)$$

Par contre, on n'a plus un simple modèle multinomial sur les variables cachées mais un champ de Markov qui permet d'instaurer des liaisons locales de voisinage :

$$P(\mathbf{Z}|\phi) = \frac{exp(-U(\mathbf{Z}|\phi))}{\sum_{\mathbf{z}\in\mathcal{Z}^I} exp(-U(\mathbf{z}|\phi))}$$

La fonction U est appelée fonction d'énergie, c'est une somme de fonctions de potentiel sur chaque clique. La communauté du traitement d'images utilise couramment le model de Potts :

$$U(\mathbf{z}|\phi) = -\beta \sum_i \sum_{j\sim i} h_{ij} \mathbb{1}_{\{z_i=z_j\}}$$

D'où le modèle de vraisemblance complète :

$$P(\mathbf{X},\mathbf{Z}|\Phi) \quad = \prod_i P(X_i|Z_i|\theta) \times \frac{exp(-U(Z_1,Z_2,\cdots,Z_I|\phi))}{\sum_{\mathbf{z}\in\mathcal{Z}^I} exp(-U(\mathbf{Z}|\phi))}$$

La résolution par algorithme EM donne la fonction Q, $\Phi = (\theta, \phi)$:

$$Q(\Phi|\Phi^n) \quad = \underbrace{\sum_{\mathbf{Z}} P(\mathbf{z}|\mathbf{x},\Phi^n) \log P(\mathbf{x}|\mathbf{z},\theta)}_{R_{\mathbf{X}}(\theta|\Phi^n)} + \underbrace{\sum_{\mathbf{Z}} P(\mathbf{z}|\mathbf{x},\Phi^n) \log P(\mathbf{z},\phi)}_{R_{\mathbf{Z}}(\phi|\Phi^n)}$$

Pour alléger la présentation, on suppose que les paramètres ϕ sont connus. Il faut seulement estimer les paramètres relatifs aux vraisemblances classifiantes mais sans connaître les affectations aux classes. On obtient après calcul :

$$R_{\mathbf{X}}(\theta|\Phi^n) = \sum_i \sum_k \log P(x_i|k,\theta) \underbrace{P(Z_i = k|\mathbf{x},\Phi^n)}_{\mu_{ik}^n}$$

On applique par exemple le principe de champ moyen. On voit alors apparaître une expression des μ_{ik}^t qui rappelle celle des modèles de mélanges. Se rajoute un terme prenant en compte la régularité spatiale autour de chaque centre, terme qui explique à lui tout seul l'intérêt de l'approche des champs de Markov cachés.

On voit que le champ de Markov permet d'intervenir sur les probabilités a posteriori en pénalisant des classes trop différentes pour des pixels D_i voisins. Ces méthodes sont relativement lourdes en pratique, mais donnent des résultats convenables en traitement d'image. La prise en compte du voisinage évite en effet un algorithme dit aveugle qui n'utilise pas le bénéfice de cette information essentielle. Ainsi, les modèles de mélanges donnent de moins bons résultats.

3.3.2 Algorithme

Dans l'algorithme stochastique de segmentation d'image précédent, on remplace de façon ad'hoc le potentiel de Potts par une fonction apte à un comportement de carte auto-organisatrice :

$U_k(z_i) = -\sum_l h(\xi_k, \xi_l) z_{il}$ où z_{il} correspond à un codage vectoriel de z_i

De cette façon, nous obtenons un lissage grâce au voisinage en favorisant des probabilités d'appartenance d'autant plus forte qu'elles seront voisines entre elles. En effet, nous cherchons à placer un document dans la classe la plus probable en cherchant des x_i optimaux ; alors il est clair que prendre des μ_{ik} le plus grand possible autour de chaque classe la plus probable fait bien diminuer les U_k.

L'algorithme proposé est alors le suivant :

Algorithme 11
Un algorithme de carte auto-organisatrice alterne les deux pas en analogie à la segmentation d'images :

Algorithme TNEM

Pas Init	*Initialisation en treillis de* θ^0,
Pas Estimation	*A résoudre par réinjection jusqu'a convergence :* $\mu_{ik}^n \approx \frac{P(x_i\|k,\theta^n)exp(\beta \sum_{l\sim k} h_{kl}\mu_{il}^n)}{\sum_m P(x_i\|m,\theta^n)exp(\beta \sum_{l\sim m} h_{ml}\mu_{il}^n)}$,
Pas Maximisation	$\theta^{n+1} = argmax_\theta \sum_i \sum_k \mu_{ik}^n P(x_i\|k,\theta)$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$, *sinon retour au pas Affectation.*

Nous faisons remarquer que l'initialisation est cruciale ici, sous peine d'aboutir à une carte non organisée. Cette constatation sera développée au niveau de la partie test.

3.3.3 Critères implicite et dérivé

L'algorithme précédent est une adaptation des algorithmes de segmentation d'image à la problématique de représentation cartographique de vecteurs multidimensionnels. Il n'est pas étonnant qu'il partage avec eux certaines propriétés. Ainsi, nous proposons ci-dessous un critère minimisé par l'algorithme. Posons μ_{ik}, des coefficient flous donnant le degré d'appartenance de x_i à C_k.

Proposition 1 *L'algorithme précédent optimise :*

$$R(\theta,\mu|\mathcal{D}) \quad = \sum_i \sum_k \mu_{ik} \log P(x_i|k;\theta) - \sum_i \sum_k \mu_{ik} \log \mu_{ik} + \frac{\beta}{2} \sum_i \mu_i^T H \mu_i$$

Preuve En effet, en introduisant un lagrangien, on a :

$$\begin{array}{ll} & R(\theta^n,\mu|\mathcal{D}) = \sum_i \sum_k \mu_{ik} \log P(x_i|k;\theta^n) - \sum_i \sum_k \mu_{ik} \log \mu_{ik} + \frac{\beta}{2} \sum_i \sum_k \sum_l h_{kl}\mu_{ik}\mu_{il} \\ \Rightarrow & \frac{\partial R(\theta^n,\mu|\mathcal{D}) - \lambda_k(\sum_k \mu_{ik} - 1)}{\partial \mu_{ik}} = \log P(x_i|k;\theta^n) - \log \mu_{ik} - 1 + \beta \sum_l h_{kl}\mu_{il} - \lambda_k \\ \Rightarrow & \mu_{ik}^n = \frac{P(x_i|k;\theta^n)exp(\beta \sum_l h_{kl}\mu_{il}^n)}{\sum_m P(x_i|m;\theta^n)exp(\beta \sum_l h_{ml}\mu_{il}^n)} \end{array}$$

Le théorème ci-après montre la convergence d'une procédure de point fixe, $\mu = F(\mu)$. Il s'agit du pas *Estimation* de l'algorithme. Le pas *Maximisation* effectue une optimisation du critère pour les paramètres des lois du mélange de distributions. □

Il s'agit d'un critère assimilable à une vraisemblance à laquelle est ajoutée un terme de régularisation d'autant plus grand que les μ_{ik} voisins sur la grille sont proches en valeurs ; donc de façon intuitive, il s'agit bien d'un algorithme permettant l'auto-organisation d'une carte. Ce critère permet de traiter plus facilement la question de convergence de l'algorithme proposé. En fait, il est très similaire à l'algorithme du NEM. Le parallèle avec les champs cachés est d'ailleurs à rapprocher également de celui fait pour le NEM. La convergence va d'ailleurs se justifier de façon relativement similaire. On trouve la condition : $\exists \beta_{max} t.q. \beta < \beta_{max}$ entraîne la CV. Le critère précédent peut d'ailleurs se généraliser à une autre distance comparant les μ_{ik} avec leurs valeurs voisines sur la grille. Nous avons étudié la méthode avec le produit scalaire sur des données simulées et réelles. Nous présentons dans le cas du produit scalaire une preuve de convergence du critère vers une solution.

Proposition 2 *Si* $N_h = max_k \sum_{l \in \mathcal{N}_k} 1$, *et* $\bar{N}_h = max_k \sum_{l \in \bar{\mathcal{N}}_k} 1$ *alors le critère pénalisé admet une unique solution, et la fonction contractante admet un point fixe solution, pour au moins* $\beta < max((KN_h - (K - \bar{N}_h)^2 + N_h \bar{N}_h)^{-1}, max_k \frac{1}{\sum_l h_{kl}})$.

Preuve On introduit un lagrangien λ_k, si bien que :

$$\frac{\partial [R(\theta, \mu|\mathcal{D}) + \lambda_k \left(\sum_k \mu_{ik} - 1\right)]}{\partial \mu_{ik}} = \log P(x_i|k, \theta) - \log \mu_{ik} - 1 + \beta \sum_l h_{kl} \mu_{il} + \lambda_k$$

On en déduit l'algorithme de point fixe que vérifie les μ_{ik} solutions. On pose $\mu = (\mu_{11}, \mu_{1,2}, \cdots, \mu_{iK}, \mu_{21}, \cdots, \mu_{IK})^T$, et F, la fonction de point fixe telle que $\mu = F(\mu)$. On vérifie avant d'étudier F si R admet bien une solution en évaluant la hessienne et son signe. Or,

$$\frac{\partial R(\theta, \mu|\mathcal{D})}{\partial \mu_{ik} \partial \mu_{jl}} = -\delta(k = l)\delta(j = i)\frac{1}{\mu_{ik}} + \beta \delta(j = i) h_{kl} \mu_{il}$$

Donc, pour toute valeur propre λ de la hessienne de R, on sait que[75] :

$$|\lambda - \frac{\partial R(\theta, \mu|\mathcal{D})}{\partial \mu_{ik} \partial \mu_{jl}}_{ik;ik}| < \sum_{jl \neq ik} |\frac{\partial R(\theta, \mu|\mathcal{D})}{\partial \mu_{ik} \partial \mu_{jl}}_{jl;jl}| \Rightarrow |\lambda + \frac{1}{\mu_{ik}}| \leq \beta \sum_l h_{kl}$$

En effet, $h_{kk} = 0$, par définition, et $\mu_{ik} \leq 1$. On en déduit que les valeurs propres de la hessienne sont bien négatives, pour $\beta < max_k \frac{1}{\sum_l h_{kl}}$, soit $\beta < N_h^{-1}$. en définitive, $R(\theta, \mu|\mathcal{D})$ admet une unique solution μ. On montre ensuite que la fonction F est contractante sous certaines conditions, si bien que la solution obtenue est celle de R. En effet :

$$\frac{\partial F_{ik}(\mu)}{\partial \mu_{jl}} = \delta(j = i)\ \beta \mu_{ik} \left(h_{kl} - \sum_m h_{ml} \mu_{im}\right)$$

Donc, on a :

$$\begin{aligned}
||F'(\mu)||_\infty &= max_{ik} \ \sum_{jl} |\tfrac{\partial F_{ik}(\mu)}{\partial \mu_{jl}}| \\
&= max_{ik} \ \beta\mu_{ik} \ \sum_l |h_{kl} - \sum_m h_{ml}\mu_{im}| \\
&= max_{ik} \ \beta\mu_{ik} \ \sum_l |\sum_m (h_{kl} - h_{ml})\mu_{im}| \\
&< max_k \ \beta \ \sum_l \sum_m |h_{kl} - h_{ml}| \\
&< max_k \ \beta \ \left(\sum_{l\in\mathcal{N}_k} \sum_m |1 - h_{ml}| + \sum_{l\in\bar{\mathcal{N}}_k} \sum_m |0 - h_{ml}| \right) \\
&< max_k \ \beta \ \left(\sum_{l\in\mathcal{N}_k} \sum_m 1 - \sum_{l\in\mathcal{N}_k} \sum_m h_{ml} + \sum_{l\in\bar{\mathcal{N}}_k} \sum_m h_{ml} \right) \\
&< \beta \ \left(KN_h - (K - \bar{N}_h)^2 + N_h\bar{N}_h \right)
\end{aligned}$$

□

3.3.4 Echec d'une résolution (sans entropie) naïve et perspective

On peut chercher à optimiser le critère en supprimant le terme d'entropie. En effet, ce terme est responsable dans l'EM des affectations aux classes non binaires, et qui oblige à calculer les μ_{ik} comme des probabilités a posteriori, au lieu de prendre une décision binaire comme le CEM. En résolvant directement, avec hypothèse de somme à l'unité sur les vecteurs μ_i, on obtient en calculant le lagrangien, en écrivant $L_i^\theta = (\log P(x_i|1,\theta), \log P(x_i|2,\theta), \cdots, \log P(x_i|K,\theta))$:

$$\mu_i = \frac{1}{\beta} \ [H^{-1}L_i^\theta - \frac{\beta + \mathbf{1}_K^T H^{-1} L_i^\theta}{\mathbf{1}_K^T H^{-1} \mathbf{1}_K} \ \mathbf{1}_K] \ \in \ [-\infty; +\infty]^K$$

où l'intervalle $[-\infty; +\infty]$ est obtenu par simulation. Or, la résolution ne prend pas en compte ici la contrainte de positivité sur les μ_i. En pratique, on obtient des valeurs parfois négatives, parfois fort élevées, bien que sommant à l'unité. En recherchant une solution d'optimisation non-linéaire standard (Uzawa), nous n'avons pas abouti à une solution convergente. Cette voie nécessite d'être explorée davantage pour aboutir à un algorithme de type CEM contraint, éventuellement de complexité moindre que l'algorithme proposé. Une solution alternative est de simplement résoudre par le TNEM, en faisant lentement décroître la température jusqu'à parvenir à un partitionnement en classes. Le recuit déterministe en résultant doit d'ailleurs permettre d'obtenir une solution convenable.

3.3.5 Conclusion sur le TNEM

Nous avons proposé un nouveau formalisme pour construire des cartes auto-organisatrices. Au lieu d'utiliser la fonction de voisinage habituelle, nous modélisons le voisinage en modifiant directement une distribution de Gibbs. A notre connaissance, le traitement de la projection en carte bidimensionnelle par des méthodes de champ de Markov est une approche originale. Elle devrait apporter des éléments de réponse à des problématiques peu abordées dans la littérature : taille de la carte, dépendances

statistiques des classes, qualité de la carte. Nous espérons que ce nouveau point de vue peut également offrir une meilleure compréhension des phénomènes en jeu dans l'auto-organisation d'une carte sémantique afin d'améliorer les résultats obtenus. Ce point est essentiel lorsque l'on cherche à calculer une projection pour un grand nombre de données de manière automatique. Obtenir une carte interprétable et résumant au mieux les données, donc la moins déformée possible est une priorité. Le modèle présenté ci-dessus permet partiellement de réduire ce problème et pose de nouvelles questions. Il faudrait notamment développer un algorithme de calcul automatique pour le facteur β et proposer des extension pour de plus grands volumes de données.

Le critère demeure flou malgré le modèle probabiliste de champ de Markov dont il dérive. Une écriture sous la forme d'un champ de Markov par une normalisation du critère semble ici inutile; en effet, l'expression obtenue paraît difficilement exploitable. Une modélisation entièrement probabiliste, non obtenue ici, apporterait vraisemblablement un modèle encore plus flexible et apte à des estimations et extensions stochastiques beaucoup plus efficaces. C'est pourquoi nous proposons le modèle suivant, le seul proposé dans ce document qui amène à une solution explicite totalement probabiliste. A notre connaissance, ce modèle que nous appelons CABSOM, est original pour les distributions discrètes.

3.4 CABSOM, version bayésienne de CASOM

Pour segmenter les lignes d'un tableau de contingence, un K-means muni d'une distance adéquate, par exemple la distance du χ^2 convient. Pour obtenir un critère de classification floue (soft), on ajoute également un terme de somme d'entropies pondérée par un coefficient T de température, comme le définit Rose dans [19]. On montrera comment ce terme permet d'interpréter le critère alterné des nuées dynamiques, usuellement utilisé pour optimiser un K-means, en un algorithme EM maximisant une vraisemblance à définir. Enfin, on ajoute un terme de régularisation pénalisant à la manière du BSOM, et amenant une contrainte d'auto-organisation des centres de classe.

3.4.1 Définition

On note $m(j) = (m_{1j}, \cdots, m_{Kj})$, car la régularisation (géométrique) s'effectue ici sur chacune des composantes indépendamment des autres, et $\mathbf{m} = (m_1, m_2, \cdots, m_K) = \theta$. Voici finalement le critère comme on le propose, avec $T > 0$, et $\gamma > 0$:

$$Q_{\chi^2}(\theta, \mu|\mathcal{D}) = \sum_i f_i \sum_k \mu_{ik} ||x_i - m_k||^2_{\chi^2} + T \sum_i \sum_k \mu_{ik} log \mu_{ik} + \gamma \sum_j ||Lm(j)||^2$$

On a choisi une distance du χ^2 afin de ne pas donner trop de poids aux colonnes sur représentées. Il ne faut pas oublier que puisque l'on travaille sur des profils lignes, on a la contrainte de somme à l'unité, que l'on va d'ailleurs également imposer aux m_k dans la suite.

3.4.2 Optimisation du critère

On alterne à la façon des nuées dynamiques un calcul des coefficients (ici flous) d'appartenance (pas Expectation) aux classes et un pas de minimisation (pas Minimisation). La convergence est assurée puisque la hessienne est strictement positive et que chaque pas ci-dessous minimise le critère. On remarque avant tout que plus un document est représenté et moins il est probable (à distance égale) dans cette expression, donc la modèle favorise d'une certaine manière les document avec peu de mots. Cette propriété remarquable n'est pas forcément avantageuse. En fait, elle va à l'encontre d'une méthode telle que l'AFC[13] qui donne plus de poids dans l'inertie décomposée aux textes dont la pondération est la plus forte, or les μ_{ik} ne sont pas seuls à pondérer les estimations. On remarque dans l'expression de la moyenne pondérée qui se substitue aux μ_{ik}, l'apparition à nouveau du facteur f_i qui cette fois-ci, contrairement au μ_{ik} précédents donne un poids maintenant plus important aux textes les plus longs (mots plus nombreux) dans les sommes pondérées. Les $m(j)$ permettent de retrouver l'ensemble des composantes des m_k. On note n l'itération à chaque pas, et les valeurs :

$$\forall i,j,k\ \bar{m}_{kj} = \frac{1}{\sum_i f_i \mu_{ik}} \sum_i f_i \mu_{ik} x_{ij} \text{ et } N = diag(\cdots, \sum_i f_i \mu_{ik}, \cdots)$$

On déduit les expressions des calculs à itérer :

Algorithme 12 *Un algorithme minimisant le critère Q_{χ^2} s'écrit :*

Algorithme pour Q_{χ^2}

Init	*Initialisation de* θ^0,
Pas E	$\forall i,k\ \mu_{ik}^n = \frac{\exp\left(-\frac{f_i}{T}\|\|x_i - m_k^n\|\|^2_{\chi^2}\right)}{\sum_l \exp\left(-\frac{f_i}{T}\|\|x_i - m_l^n\|\|^2_{\chi^2}\right)}$
Pas M	$m_{(j)}^{n+1} = (\frac{1}{f_j} N^n + \gamma L^T L)^{-1} \times$
	$\cdots \left\{ \frac{1}{f_j} N \bar{m}_{(j)} + [\sum_m (\frac{1}{f_m} N^t + \gamma L^T L)^{-1}]^{-1} \times \right.$
	$\left. \cdots [\mathbb{I}_K - \sum_{j'} \frac{1}{f_{j'}} (\frac{1}{f_{j'}} N^n + \gamma L^T L)^{-1} N^n \bar{m}^t_{(j')}] \right\}$
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
	sinon retour à 2.

L'algorithme proposé converge vers une solution du critère :

Proposition 3
L'algorithme présenté alterne deux phases qui minimisent le critère à chaque pas. Il converge vers un minimum (local) du critère.

Preuve Le premier pas diminue pour les μ_{ik}, à $\mathbf{m}$ fixé, le critère puisque la hessienne est diagonale positive. Le second pas également. Il résout le critère pour $\mathbf{m}$ avec les μ_{ik} fixés. En effet :

$$\begin{aligned}
& \frac{\partial Q_{\chi^2}(\theta,\mu|\mathcal{D})+\lambda_k(\sum_j m_{kj}-1)}{\partial m_{kj}} \\
= \quad & -2(\textstyle\sum_i f_i\mu_{ik})\frac{1}{f_j}(\bar{m}_{kj}-m_{kj})+2\gamma(L^TLm(j))_j+\lambda_k \\
\Rightarrow \quad & \text{(en passant au vecteur)} \\
& \mathbb{O}_{K,1}=-2N\tfrac{1}{f_j}\bar{m}_{(j)}+2N\tfrac{1}{f_j}m_{(j)}+2\gamma L^TLm_{(j)}+\Lambda \\
\Rightarrow \quad & \text{(pour le lagrangien divisé par +2)} \\
& m_{(j)}=(\tfrac{1}{f_j}N+L^TL)^{-1}N\bar{m}_{(j)}+(\tfrac{1}{f_j}N+L^TL)^{-1}\Lambda
\end{aligned}$$

D'où le résultat par somme en colonne sur les $m_{(j)}$, puisque l'on sait que $\sum_j m_{(j)}=\mathbb{I}_K$. □

Il est introduit un terme de température T comme le préconise Rose. Si on fait tendre T vers ∞, on obtient une équiprobabilité : le maximum des termes d'entropie. Si par contre, T tends vers 0, on tend vers une classification dure : les μ_{ik} valent 1 pour la classe la plus probable (a posteriori) et 0 ailleurs. Une façon de procéder est donc de baisser la température au fur et à mesure des itérations dans le but de ne pas tomber trop rapidement dans des extrema locaux. Intuitivement, comme les μ_{ik} sont lissés au début, l'optimisation atteint moins rapidement une solution. L'expression obtenue est relativement lourde (nombreuses inversions, produits matriciel), mais si l'on se souvient que N est diagonale, ce n'est plus le cas. On procède à une simplification de l'expression grâce à quelques calculs algébriques pour obtenir l'algorithme ci-dessous. On note SVD l'opération de diagonalisation. Un critère d'arrêt est par exemple $||\mathbf{m}^n-\mathbf{m}^{n-1}||<\epsilon$ pour un ϵ assez petit arbitraire, et une norme adéquate, par exemple, celle de Frobenius.

Algorithme 13
Un algorithme minimisant le critère Q_{χ^2}, simplifié à l'aide d'une SVD s'écrit :

Algorithme avec SVD pour Q_{χ^2}

Init	*Initialisation de* θ^0,
Pas E	$\forall i,k\ \mu_{ik}^n = \frac{\exp\left(-\frac{f_i}{T}\|\|x_i - m_k^n\|\|^2_{\chi^2}\right)}{\sum_l \exp\left(-\frac{f_i}{T}\|\|x_i - m_l^n\|\|^2_{\chi^2}\right)}$
Pas SVD	• $\forall k,j\ \bar{m}_{kj}^n$ *et* N^n,
	• $[R^n, \Sigma^n] = SVD((L(N^n)^{-1/2})^T(L(N^n)^{-1/2}))$,
	$\Sigma^n = diag(\cdots, \sigma_k^n, \cdots,)$,
	• $\forall j, S_j^n = diag(\cdots, (\frac{1}{f_j} + \gamma\sigma_k^n)^{-1}, \cdots)$,
	$S^n = diag(\cdots, (\sum_j (\frac{1}{f_j} + \gamma\sigma_k^n)^{-1})^{-1}, \cdots)$,
	• $U_n = (N^n)^{-1/2} R^n$,
Pas M	$m_{(j)}^{n+1} = [U_n S_j^n] \times \Big(\frac{1}{f_j}[U_n^T N^n]\bar{m}_{(j)} + [S^n U_n^T N^n]\mathbb{I}_k -$
	$\cdots S^n \times [\sum_{j'} \frac{1}{P_{j'}} S_{j'}^n U_n^T N^n \bar{m}_{(j')}^n]\Big)$,
Test	$\|\|\theta^n - \theta^{n-1}\|\| < \epsilon$ *alors* $\hat{\theta} = \theta^n$,
	sinon retour à 2.

Preuve
En effet, avec les notations de l'algorithme, on a :

$$
\begin{aligned}
& (\tfrac{1}{f_j}N + \gamma L^T L)^{-1} \\
= \;& N^{-1/2}(\tfrac{1}{f_j}\mathbb{I}_K + (LN^{-1/2})^T(LN^{-1/2}))^{-1}N^{-1/2} \\
= \;& US_jU^T \\
\Rightarrow & \\
& m_{(j)}^{n+1} \\
= \;& (\tfrac{1}{f_j}N^n + \gamma L^T L)^{-1}\times \\
& \cdots \left\{\tfrac{1}{f_j}N^n\bar{m}_{(j)} + [\textstyle\sum_m(\frac{1}{f_m}N^n + \gamma L^T L)^{-1}]^{-1} \times [\mathbb{I}_K - \sum_{j'}\frac{1}{f_{j'}}(\frac{1}{f_{j'}}N^n + \gamma L^T L)^{-1}N^n\bar{m}_{(j')}^n]\right\} \\
= \;& U_n S_j^n U_n^T \times \left\{\tfrac{1}{f_j}N^n\bar{m}_{(j)} + U_n S^n U_n^T N^n[\mathbb{I}_K - \textstyle\sum_{j'} U_n S_{j'}^n U_n^T N^n \bar{m}_{(j')}]\right\}
\end{aligned}
$$

□

On obtient finalement, grâce aux calculs intermédiaires des matrices U^n, une expression simplifiée rapide à calculer étant donné que la dimension des matrices est K, d'autant plus que N^n est diagonale : la matrice à diagonaliser est de taille réduite. En outre, la dernière somme peut se calculer une fois unique pour l'ensemble des $m_{(j)}$. Les précalculs précédents nécessitent en réalité un compromis entre capacité de calcul et capacité de stockage en mémoire.

3.4.3 A propos de la modélisation

En calculant $\exp(-\alpha Q_{\chi^2}(\theta, \mu|\mathcal{D}))$ (avec $T = 0$, $\alpha > 0$) il paraît difficile d'obtenir ensuite une vraisemblance simple par normalisation. Cela vient du fait que les pondérations f_i empêche la factorisation en gaussiennes. Tout au plus, on arrive à une vraisemblance :

$$\mathcal{L}(\theta|\mathcal{D}) \propto \Big[\prod_i \sum_k \exp\,(-\alpha \frac{||x_i - m_k||^2_{\chi^2}}{T/f_i})\Big] \times \Big[\prod_j exp(\alpha\gamma||Lm(j)||^2)\Big]$$

Ce modèle pose problème à cause de la variance des classes qui est fonction ici du document : on n'a pas un échantillon de données i.i.d. Ces données sont indépendantes, mais de lois différentes, donc les propriétés du maximum de vraisemblance ne s'appliquent plus de façon ad'hoc. Pourtant, d'un point de vue purement algébrique, les probabilités se normalisent ici facilement.

En outre, la positivité des solutions $\mathbf{m}$ n'est plus évidente. Pour s'en assurer, on modélise par exemple chaque vecteur de probabilité par une exponentielle normée. On note $m_k(\beta) = \frac{exp(\beta_{kj})}{\sum_l exp(\beta_{kl})}$, $\beta_j = (\beta_{1j}, \beta_{2j}, \cdots, \beta_{Kj})^T$, d'où :

$$\begin{aligned}\mathcal{L}(\theta|\mathcal{D}) &= P(\mathcal{D}|\beta) \times P(\beta) \\ &\propto \Big[\textstyle\prod_i \sum_k \exp\,(-\alpha \frac{||x_i - m_k(\beta)||^2_{\chi^2}}{T/f_i})\Big] \times \Big[\prod_j exp(\alpha\gamma||L\beta_j||^2)\Big]\end{aligned}$$

Cette approche a le désavantage de nécessiter une optimisation non analytique. Au lieu d'un algorithme EM, où le pas d'optimisation est explicite, il devient nécessaire de recourir à des techniques numériques, type Newton (EM généralisé). En modifiant la distance du χ^2 en l'euclidienne usuelle pour des vecteurs x_i à J valeurs dans R, la factorisation devient possible, et on retrouve le BSOM d'Utsugi[57] (pour T=1). On a d'ailleurs noté les itérations Expectation-Minimisation pour souligner l'analogie avec l'algorithme EM (Expectation-Maximization) calculé sur le BSOM.

Enfin, on aurait pu partir directement de ce modèle en choisissant une distribution de classe multinomial. Or, du fait de la présente régularisation, on arrive également à une solution non solvable analytiquement. Ce modèle s'écrit , avec $P^{\beta}_{j|k} = \frac{exp(\beta_{kj})}{\sum_l exp(\beta_{kl})}$:

$$\begin{aligned}\mathcal{L}(\theta|\mathcal{D}) &= P(\mathcal{D}|\beta) \times P(\beta) \\ &\propto \Big[\textstyle\prod_i \sum_k \prod_j {P^{\beta}_{j|k}}^{N_{ij}}\Big] \times \Big[\prod_j exp(\gamma||L\beta_j||^2)\Big]\end{aligned}$$

Utsugi a montré en quoi le SOM peut se considérer comme une approximation du BSOM. Ici, le modèle proposé étant très proche du BSOM, on a une similitude comparable pour des données différentes. Par contre, l'expression de la minimisation rend a priori difficile une comparaison directe des algorithmes, en raison du facteur de normalisation des vecteurs solutions.

3.4.4 Conclusion sur CABSOM

Cette nouvelle méthode que l'on nomme CABSOM, pour une cartographie des tableaux de contingence a l'intérêt d'effectuer des calculs matriciels de taille K ; elle évite

des calculs coûteux pour des matrices de grandes tailles. Le résultat est en définitive un peu plus lourd que le cas gaussien du BSOM, et les calculs peuvent être discutables du point de vue de la stabilité numérique en raison des calculs d'inverse et de la SVD puisqu'au final, on veut aboutir à des vecteurs de probabilité. De nombreuses questions se posent encore à propos de ce genre de critère, notamment à propos des estimations des données non i.i.d., et de la détermination du facteur γ optimal, de la positivité, et de la stabilité. La forme de la matrice L paraît également un facteur important dans la réussite de l'auto-organisation, et sur laquelle il reste à travailler. L'algorithme a le mérite de mettre en évidence le genre de problèmes auxquels l'on peut être confronté lorsque l'on passe de données dans $\mathcal{R}$ à des données dans $[0, 1]$.

Chapitre 4

Simulations et applications aux matrices textuelles

4.1 Test sur données circulaires

On génère des données approximativement circulaires : des points équidistribués sur un cercle de rayon 0.2 et centre (0.5,0.5), et on prend la matrice de contingence correspondant dont les lignes somment approximativement à 10. Cela permet d'introduire du bruit au niveau des profils-lignes. On peut par exemple utiliser le programme MATLAB suivant :

```
%Génération cercle discret.
U=1:2*pi/I:2*pi;
X = (0.5 + 0.2 * cos(U))^2;
Y = (0.5 + 0.2 * sin(U))^2;
Z = 1 - X - Y;
D = round(10 .* [X' Y' Z']);
```

L'intérêt de telles données est leur représentation dans une espace à 3 dimensions par les profils-lignes. Nous allons donc observer différentes estimations des paramètres à partir des algorithmes proposés. On peut en effet comparer visuellement les résultats des méthodes. La présence d'un biais éventuel se vérifie aisément en raison de la forme circulaire, de façon empirique en confrontant les rayons estimé et réel. On procède aux tests avec les algorithmes pour ce cas en dimensions J=3 : les itérations 1, 2, 3, 4, 5, 15, 30, 60 apparaissent sur la figure 4.1. On a tracé en pointillé le vrai cercle et par des croix les points de l'échantillon D ; on remarque sur cet exemple une légère sous estimation du rayon pour les méthodes alternatives au CASOM. La forme circulaire est bien retrouvée : visuellement on peut vérifier que l'on retrouve bien la structure des données. Cependant, les deux algorithmes alternatifs demandent un réglage précis des paramètres. En outre, nous avons du introduire systématiquement une température assez faible afin d'augmenter suffisamment le rayon du cercle estimé.

4.1.1 Cas CASOM

La figure 4.1 montre un exemple d'exécution pour CASOM. On constate en effet une propriété d'auto-organisation des centres de classes qui forment une discrétisation du cercle. Par contre, il apparaît un sur-apprentissage ou taux d'erreur croissant pour un éventuel échantillon de test.

En effet, le lissage disparaît en fin d'estimation et la forme circulaire a tendance à être remplacée par une ligne en dent de scie. Autrement dit, il pourrait être intéressant de procéder à de la validation croisée pour éviter de trop spécialiser la carte et aboutir à une classification trop bonne au préjudice de la structure distributionnelle. Une autre

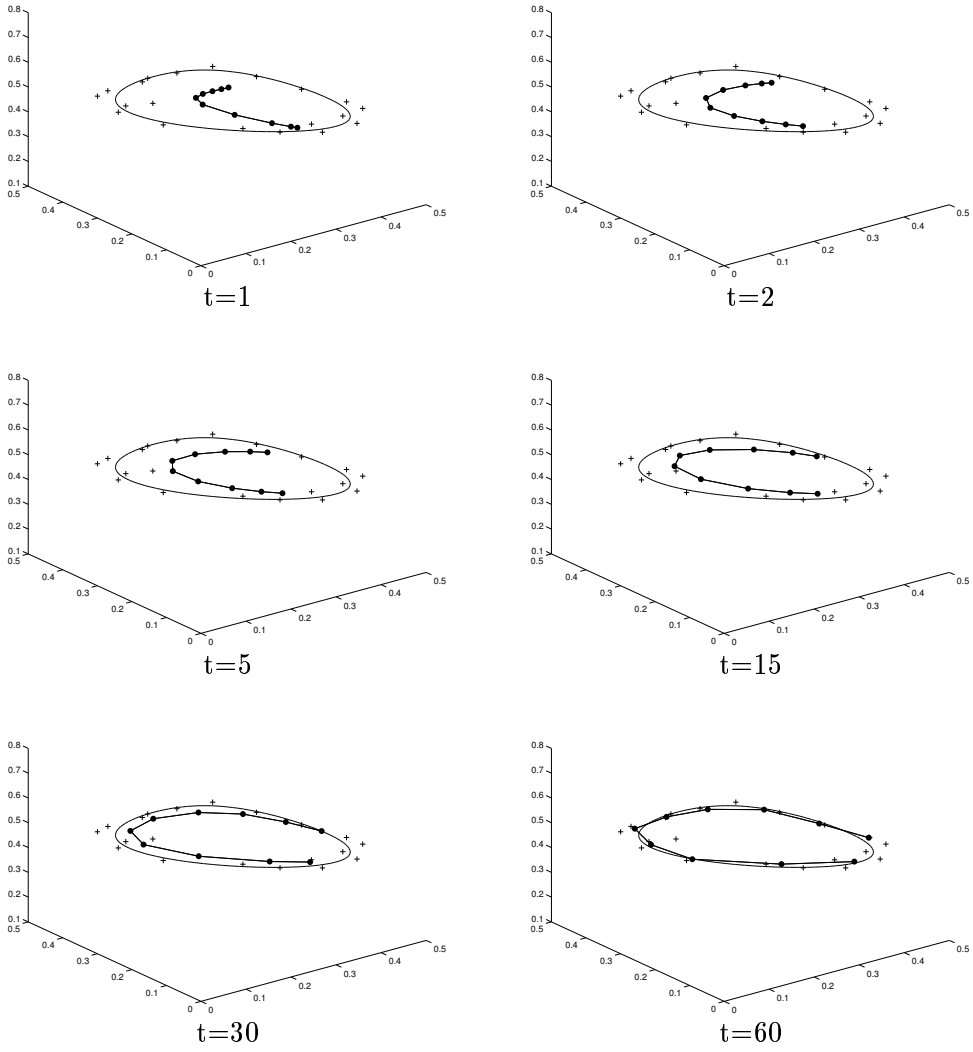


FIG. 4.1 – Algorithme testé sur des données circulaires : cas CASOM

solution serait d'utiliser un autre algorithme à la suite, mais qui lui possède un lissage réglable, à la manière des deux algorithmes suivants (CABSOM et TNEM).

4.1.2 Cas CABSOM

Pour cet algorithme, figure 4.2 un biais sur le rayon apparaît clairement en fonction du facteur lissant et de la température : nous avons mis une température très faible, de l'ordre du millième ; nous aurions pû le corriger manuellement pour montrer une meilleure solution. Par contre, ce constat privilégie l'hypothèse empirique d'une solution à valeurs dans $[0,1]^J$, puisque le biais entraîne des valeurs rétractées vers le centre de gravité du nuage.

Il faut donc ici à la fois régler la température, et le facteur lissant pour augmenter le rayon obtenu. Plus le facteur est faible, plus le rayon est grand : plus le critère de classification prime sur la pénalisation, mais alors, des minimum locaux apparaissent plus probables en pratique. Il faudrait essayer de procéder à une réduction lente du facteur lissant à la manière d'un recuit.

4.1.3 Cas TNEM

De même que pour le cas suivant, figure 4.3 cette méthode présente un biais important pour un mauvais réglage des paramètres : température essentiellement.

De plus, nous n'avons pas constaté de très forte propriété auto-organisatrice : on n'observe pas de sortie systématique d'un minimum local pour une topologie de la grille très déformée, même au début de l'algorithme si la température est trop faible. On se retrouve confronté à deux propriétés opposées : une température trop élevée permet une certaine auto-organisation, mais entraîne un biais important sur le rayon, donc une mauvaise classification. Une température plus basse, rendant les affectations davantage binaires altère l'auto-organisation, mais augmente le rayon. Donc un recuit à l'aide de la température paraît la meilleure manière de régler les paramètres. Dans les cas bien initialisés, on peut également constater la vitesse (en terme du nombre de pas) de l'algorithme qui trouve quasiment immédiatement la bonne forme circulaire. La question de vitesse en nombre de pas avant la convergence nous est apparu comme une problématique intéressante bien que guère abordée dans la littérature. Cela doit d'ailleurs expliquer la propension de cette méthode à tomber dans les minima locaux. On préconise donc cet algorithme par exemple en sortie de CASOM, pour en quelque sorte lisser la solution si jamais la classification était trop avancée. On remarque également numériquement les problèmes, même sur ce petit exemple de normalisation des coefficients d'affectation aux classes, lorsque la taille des documents synthétiques augmente. En définitive, on peut penser que l'estimation des cartes auto-organisatrices est finalement un problème comparable à la segmentation d'image bien qu'ici nous ne disposions pas d'information a priori sur le voisinage des données d'entrée. On peut en outre constater des phénomènes cycliques : après un certain nombre de pas , deux

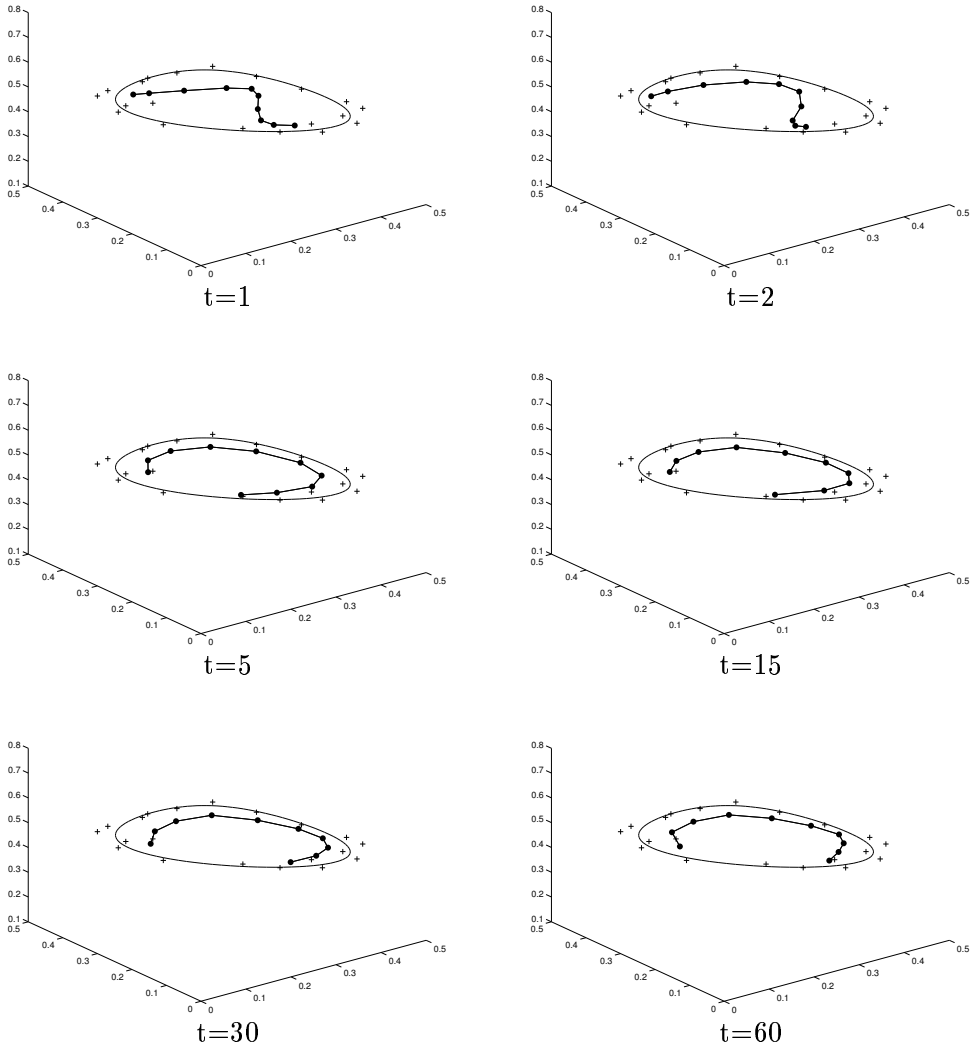


FIG. 4.2 – Algorithme testé sur des données circulaires : cas CABSOM.

estimations reviennent l'une après l'autre. Une valeur de β proche de la limite de CV nous est apparue comme une bonne valeur. Toutes ces remarques demanderaient de plus amples investigations afin de les valider de façon certaine.

4.1.4 Simulations Monte-Carlo

4.1.4.1 Cas CASOM

Tableau des indicateurs

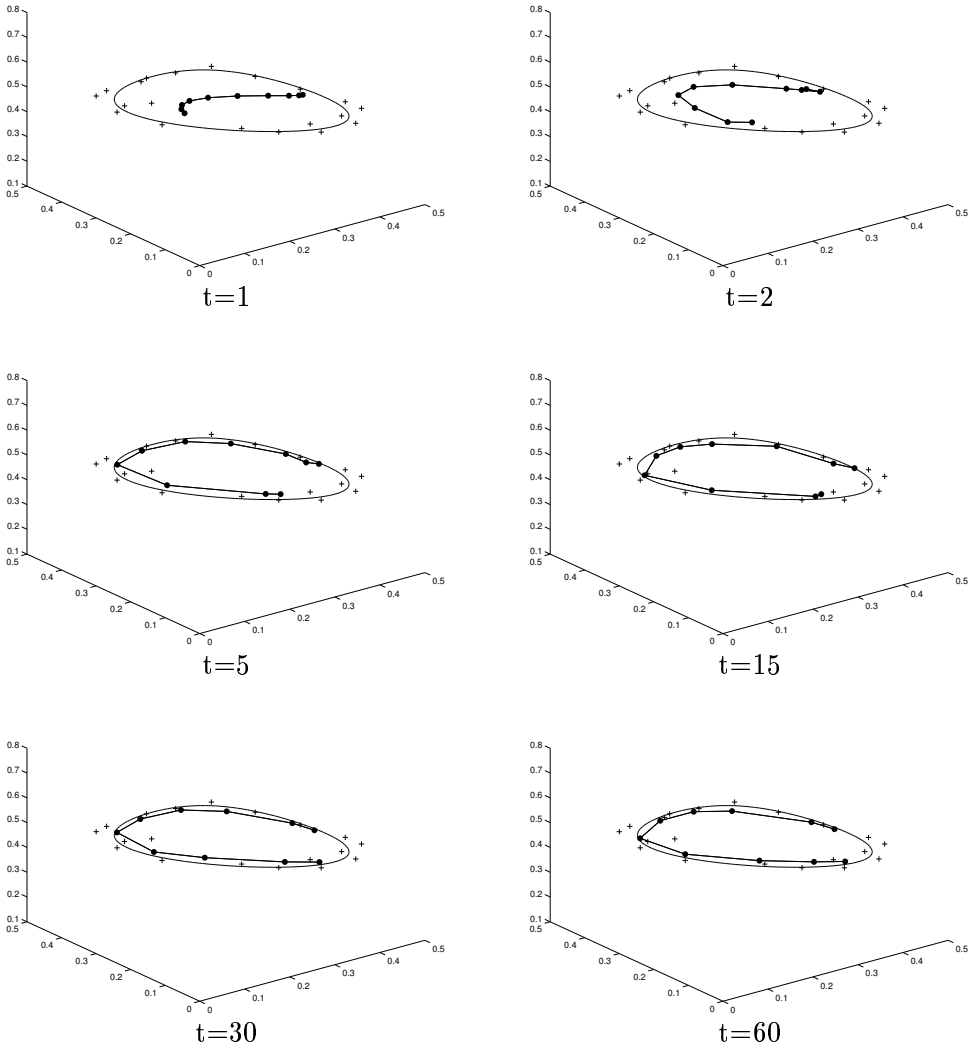


FIG. 4.3 – Algorithme testé sur des données circulaires : cas TNEM.

K	e(IS)	s(IS)	e(C)	s(C)	e(KL)	s(KL)	e(KLE)	s(KLE)	e(MKLE)	s(MKLE)
2	2,01	0,00	0,27	0,00	60,54	0,00	1715,67	0,00	315,82	0,00
3	2,92	0,00	0,47	0,00	36,18	0,00	1152,40	0,00	320,67	0,00
4	3,87	0,00	0,72	0,00	24,51	0,00	788,55	0,00	241,41	0,00
5	4,72	0,03	1,07	0,02	15,72	0,06	663,68	96,22	254,21	48,06
6	5,74	0,02	1,33	0,02	11,10	0,28	492,35	40,98	191,66	26,16
7	6,63	0,02	1,56	0,01	7,27	0,30	389,46	45,87	184,36	48,01
8	7,59	0,02	1,83	0,02	6,50	0,75	299,59	17,59	119,75	19,60
9	8,52	0,06	2,06	0,07	4,97	0,67	244,92	37,51	105,42	18,77
10	9,44	0,04	2,31	0,06	3,80	0,74	218,68	25,04	96,37	23,98
11	10,45	0,07	2,56	0,11	2,58	0,64	156,89	20,23	86,50	11,36
12	11,42	0,12	2,76	0,13	2,30	0,51	148,79	23,24	83,05	16,09
13	12,29	0,15	3,08	0,13	1,34	0,47	105,05	17,05	66,36	8,44
14	13,29	0,07	3,26	0,10	0,87	0,37	91,56	21,12	66,08	14,82
15	14,32	0,11	3,40	0,12	0,79	0,35	90,74	26,50	66,46	18,57

On calcule pour K=2,3, ..., 15, un ensemble de 10 solutions CASOM sur les données circulaires précédentes. Cette simulation permet d'évaluer la moyenne et écart-type des critères des divers indicateurs dans le tableau ci-contre. Dans ce tableau les indicateurs IS, C, KL, KLE, et MKLE, sont respectivement l'entropie totale, le χ^2, le critère de divergence-intra (/2), le critère de divergence-intra étendu au plus proches voisins (/2), et sa moyenne asymptotique(/2). En fait, les valeurs calculées ne sont pas les divergences, mais les distances du χ^2 correspondantes (d'où la division par 2, pour l'approximation). Les notations e() et s() indiquent les moyennes et écart-types empiriques obtenus à partir des 10 simulations par nombre de classes. Avant toute chose, on constate la croissance quasi-linéaire de l'entropie totale et du χ^2 du tableau sous-jacent aux multinomiales. Ensuite, un petit nombre de classes aboutit a peu de solutions différentes contrairement à un plus grand nombre. Les autres indicateurs sont décroissants, indiquant juste que les données sont de plus en plus proches des centres considérés, puisqu'ils sont plus nombreux : on parle de sur-apprentissage. On voit en outre que KLE et MKLE deviennent du même ordre de grandeur pour des valeurs élevées de K.

Vers un choix de modèle pour CASOM

On trace la divergence intra (version distance χ^2 que l'on multiplie par la normalisation $N_{\bullet\bullet}/I$ afin de comparer la valeur à J-1 = 2). Le critère permet de décider K=8, comme une bonne valeur minimal de nombre de classes. En effet, l'écart-type d'un test asymptotique à 95% vaut ici 0.28 environ. On remarque que cette valeur est justement exactement l'endroit où le modèle devient plus sensible à l'initialisation (variance élevée de KL).

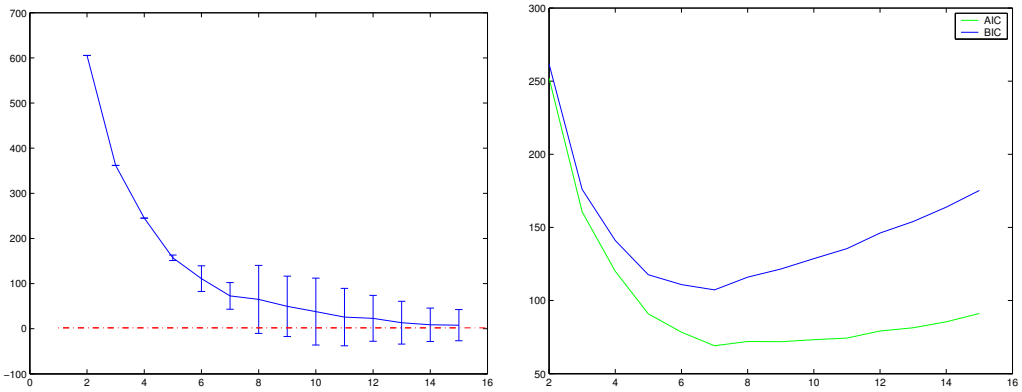


FIG. 4.4 – Critère de choix de modèle asymptotique classique et bayésien heuristique. Un test fréquentiste est visible sur la figure de gauche. En comparant à J-1 (ligne horizontale), on aboutit à un ordre de valeur pour le nombre de classes. Le test heuristique bayésien apparaît à la droite pour AIC et BIC. Effectué à partir de la valeur moyenne de KL, il s'agit de choisir K pour le minimum des courbes.

D'ailleurs, si on utilise de façon heuristique un critère bayésien, on aboutit à une valeur comparable, mais différente. On parle de critère AIC, et BIC. On calcule les courbes AIC = 2 KL + 2 (3 K -1) et BIC = 2 KL + log(IS) (3 K -1) ; on déduit ici à partir de ces critères qu'une bonne valeur est K=7, donc inférieure au choix précédent, mais relativement proche. Cet exemple simulé montre seulement les perspectives de choix de modèle à l'aide de CASOM. Il faudrait des simulations plus poussées pour dégager des résultats significatifs. Il est clair qu'une étude analytique des critères bayésiens dans le cas CASOM apporterait des réponses plus précises. On remarque que ces critères ne prennent pas réellement en compte le spatial. Introduire cette notion paraît pourtant fondamental. Il s'agirait par exemple d'user du TNEM avec les mêmes critères, de façon tout autant, sinon plus heuristique. D'autres tests sont également envisageables.

4.1.4.2 Cas TNEM

Avant toute chose, il est clair que les critères sont moins exacts dans le cas CASOM car le TNEM, par définition, n'effectue pas exactement une classification dure des données comme le réalise CASOM, puisque le TNEM doit maximiser à la fois la logvraisemblance classifiante et le facteur pénalisant sur les coefficients d'appartenance aux classes. Les figures représentent la moyenne empirique de l'intra-divergence (fig 4.1.4.2), ainsi que sa variance empirique (fig 4.1.4.2), la moyenne empirique du critère étendu (fig 4.1.4.2), ainsi que sa variance (fig 4.1.4.2), et enfin la moyenne empirique de la moyenne limite du critére étendu (fig 4.1.4.2) et sa variance(fig 4.1.4.2).

Ensuite, il faut remarquer qu'une température basse et un lissage élevé sont propices à une bonne classification, mais également à une bonne valeur (minimisée) du critère étendu. Ensuite, la variance est très variée, ce qui permet de dire que l'initialisation pour cette méthode paraît crucial sous peine de tomber dans un minimum local très éloigné de la solution optimale. On remarque là encore l'importance des températures car des valeurs élevées entraînent une plus grande variance, et donc une plus grande dépendance à l'initialisation. Le critère de moyenne limite de la divergence étendue présente des profils de courbes différents, ce qui tend à indiquer que des températures élevées entraînent des facteurs multinomiaux identiques, ce qui semble pertinent intuitivement. La variance est également inversement plus importante pour des températures basses ; on remarque la prépondérance de l'effet de *bêta*. Ces résultats tendent à supposer l'existence de valeurs optimales pour les deux températures.

Nous avons regroupé les deux termes sous le nom *température* en raison des effets observés. D'ailleurs en traitement d'image, où la température classique (devant l'entropie des coefficients d'appartenance) n'est pas forcément introduite, le lissage fait office de température pour les opérations de segmentation. Une procédure de diminution (lente) des températures permettrait d'obtenir une classification, tout en amenant les paramètres dans la bonne configuration d'auto-organisation. Enfin, nous obtenons

là encore des croissances linéaires pour l'entropie et le χ^2 d'indépendance du tableau de facteurs multinomiaux, non montrés ici.

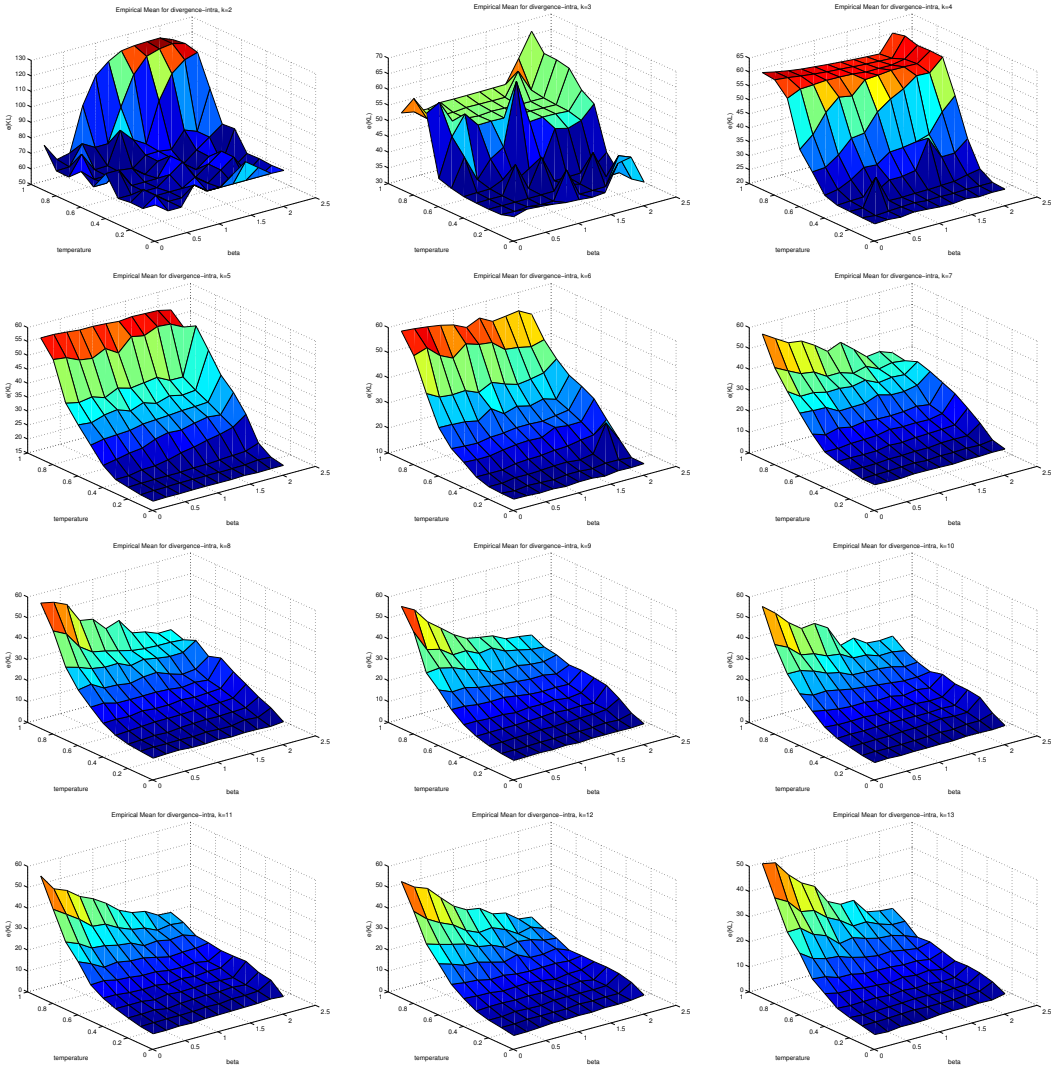


FIG. 4.5 – Moyennes empiriques de la Divergence-intra en fonction de la température et le lissage, et pour un nombre de classes fixés.

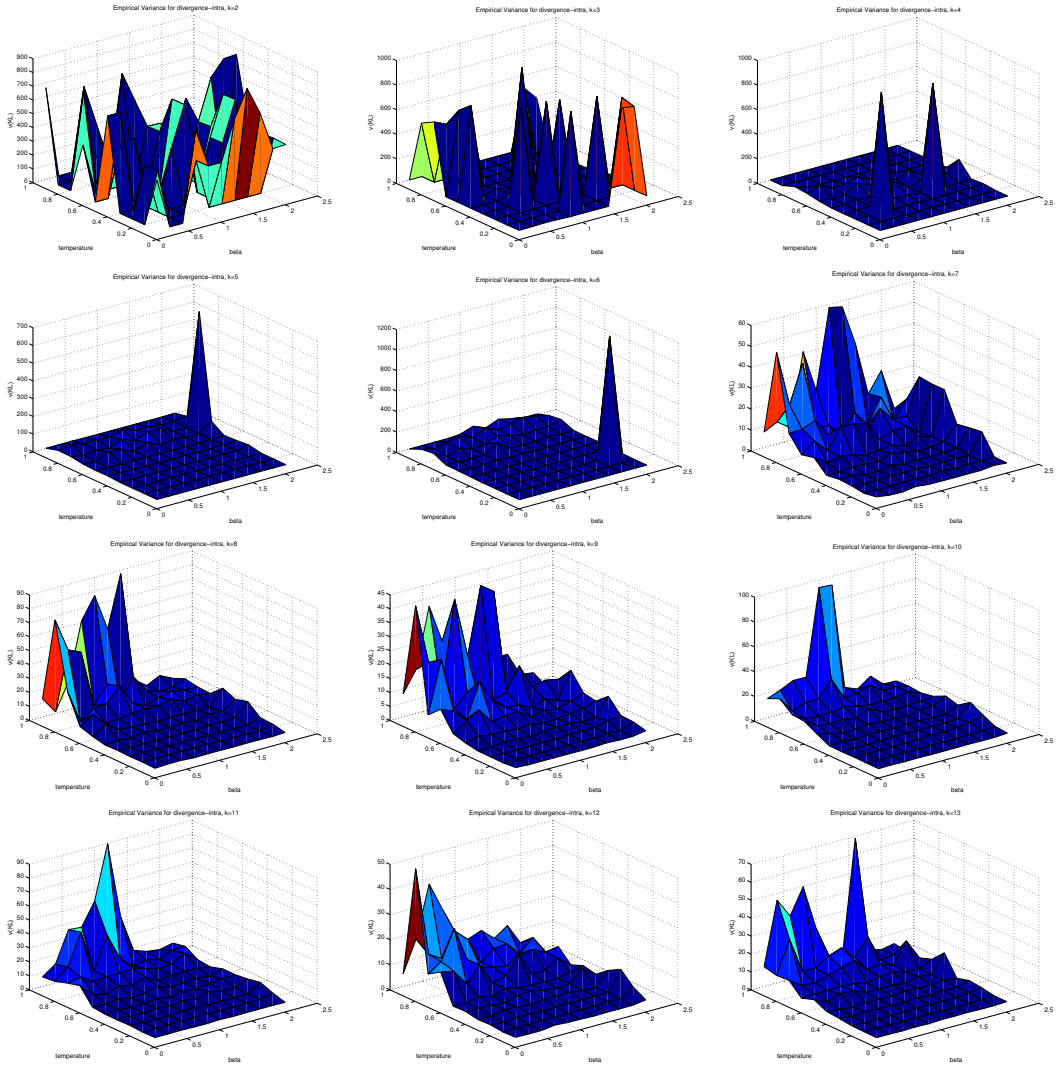


FIG. 4.6 – Variances empiriques de la Divergence-intra en fonction de la température et le lissage, et pour un nombre de classes fixés.

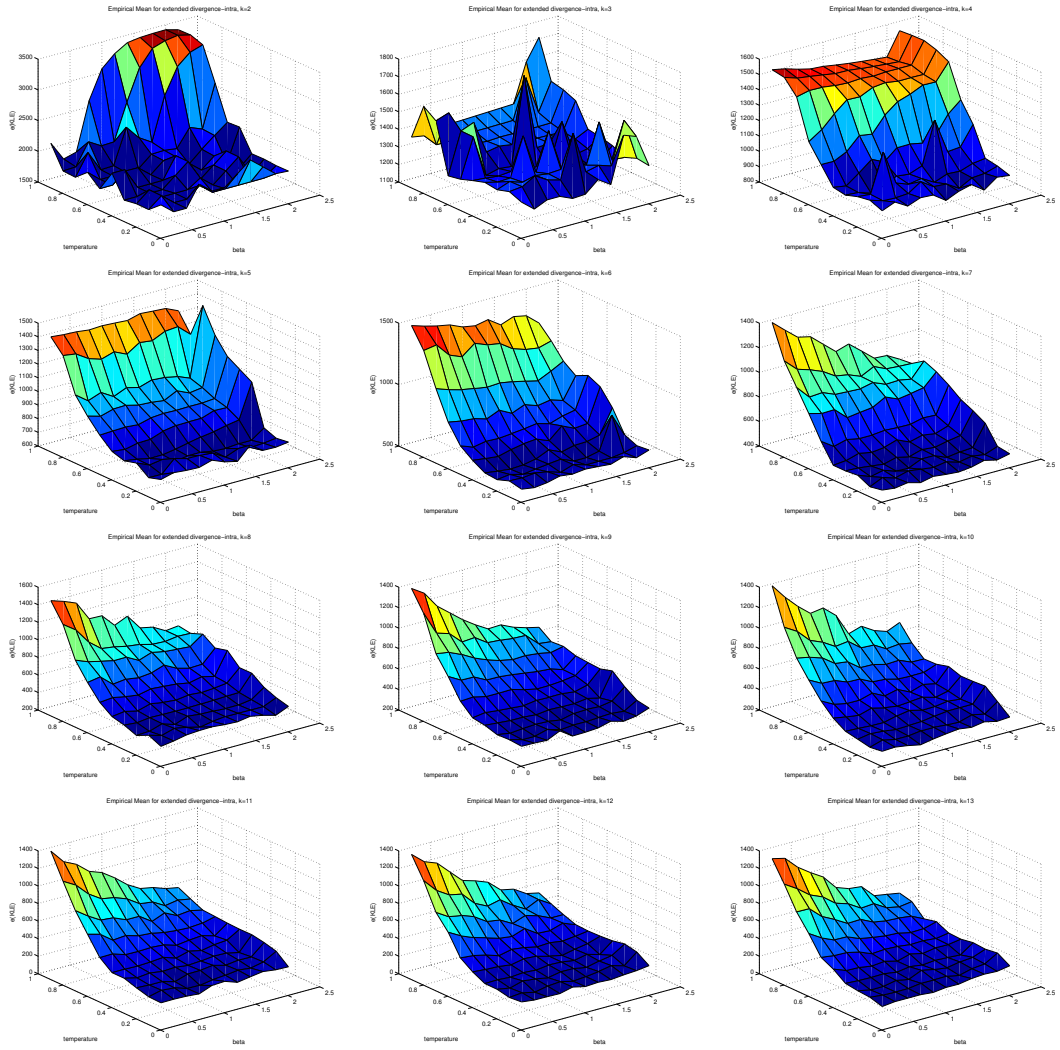


FIG. 4.7 – Moyennes empiriques de la Divergence-intra étendue en fonction de la température et le lissage, et pour un nombre de classes fixés.

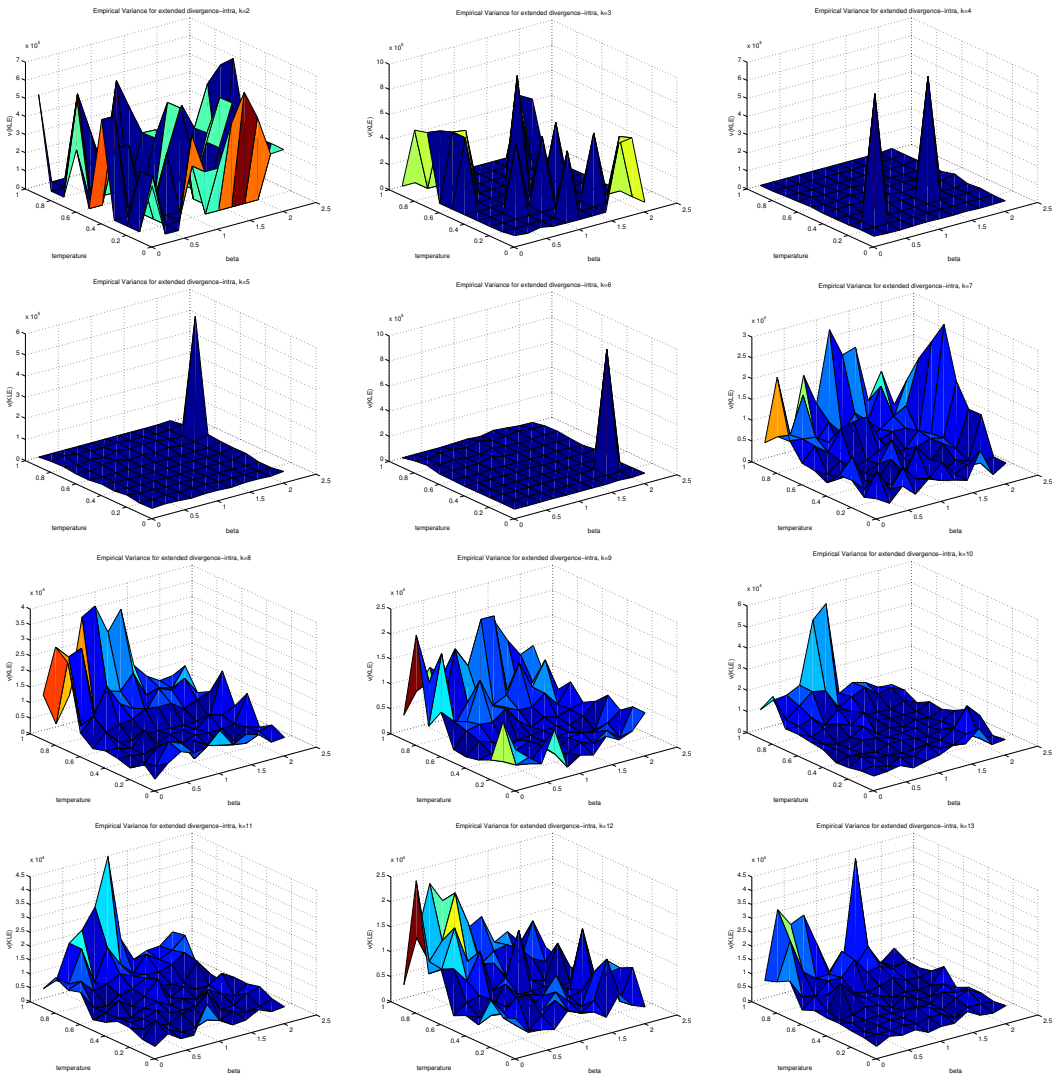


FIG. 4.8 – Variances empiriques de la Divergence-intra étendue en fonction de la température et le lissage, et pour un nombre de classes fixés.

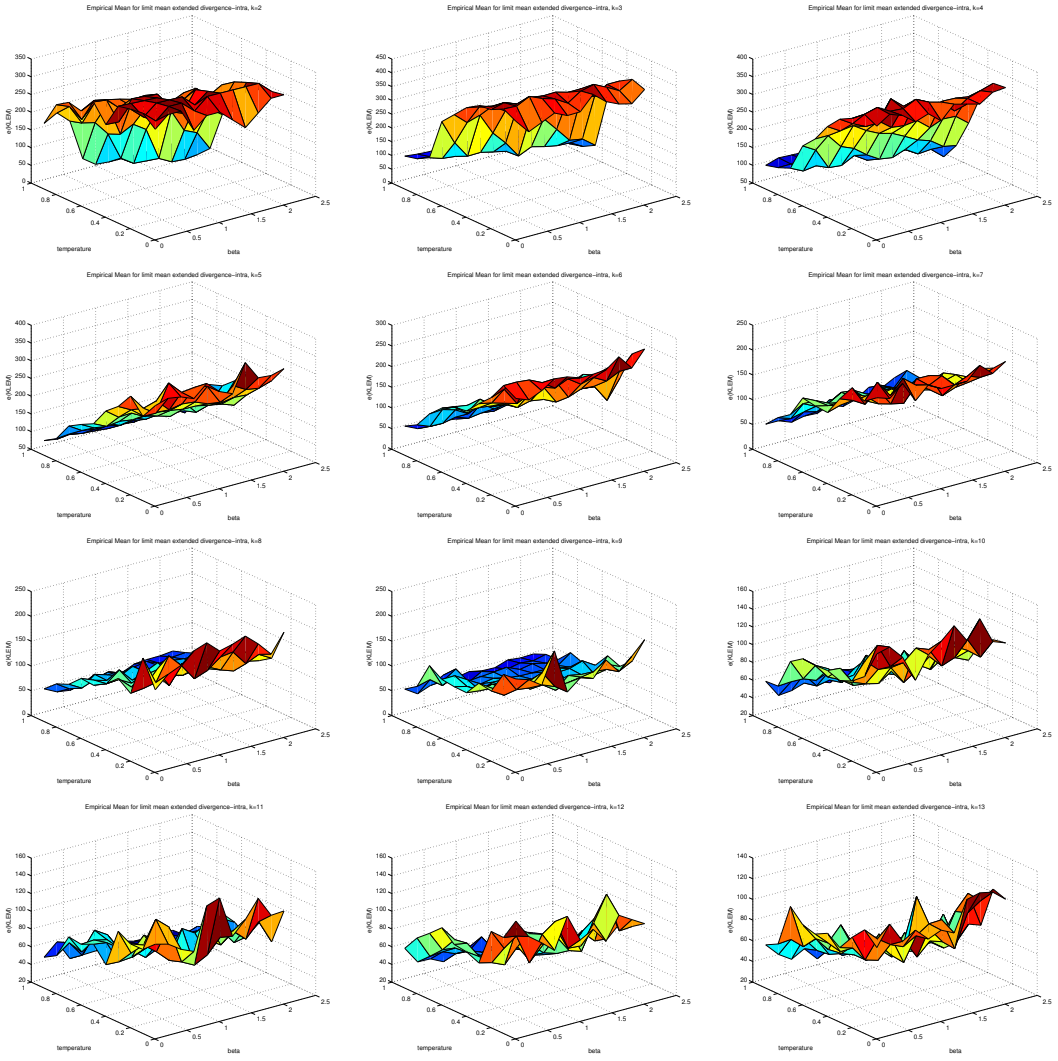


FIG. 4.9 – Moyennes empiriques de la moyenne limite de la Divergence-intra étendue en fonction de la température et le lissage, et pour un nombre de classes fixés.

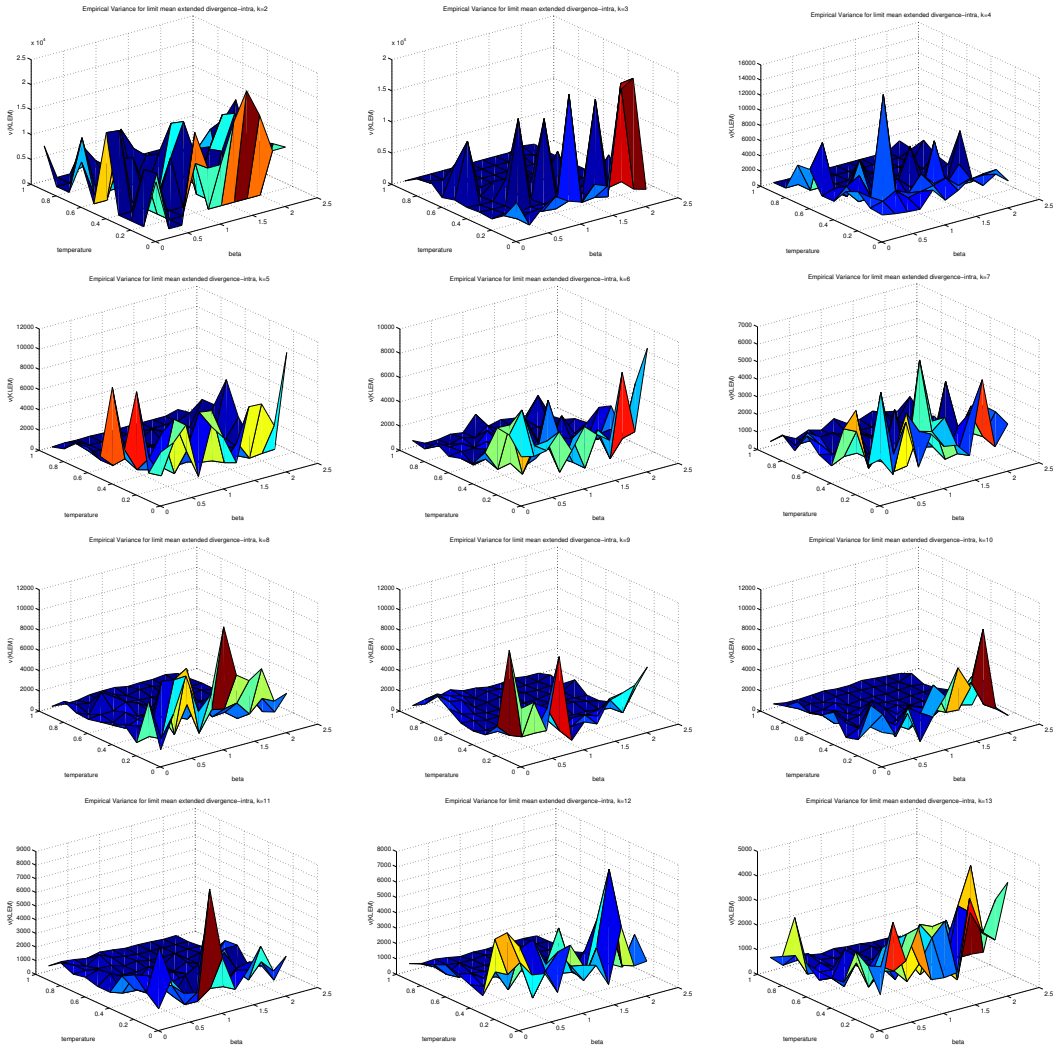


FIG. 4.10 – Variances empiriques de la moyenne limite de la Divergence-intra étendue en fonction de la température et le lissage, et pour un nombre de classes fixés.

4.2 Expériences sur les données textuelles

Les données étudiées subissent le prétraitement décrit en introduction, pour obtenir un tableau de contingence textuel. Un certain vocabulaire est choisi puis des vecteurs sont formés pour chacun des documents en comptabilisant les occurrences des termes du vocabulaire. Une fois réalisée la projection cartographique, le contenu d'un corpus entier s'explore à l'aide d'un outil de navigation comme nous allons le voir dans la suite.

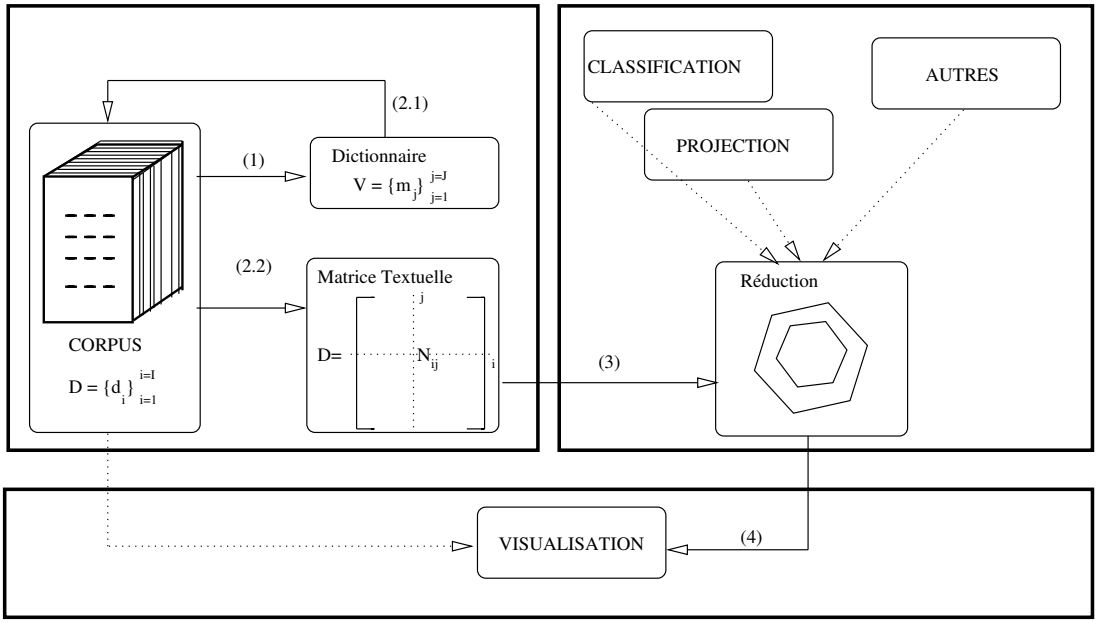


FIG. 4.11 – Schéma général du traitement d'un corpus de textes.

D'un point de vue pratique, la carte neuronale change d'apparence après chaque nouvel apprentissage si les initialisations sont différentes. Cela provient du fait que la grille de neurones est libre d'effectuer des mouvements (rotations ou symétries) dans l'espace des données, et que des minima locaux peuvent être atteints avant l'optimum. Il faut estimer plusieurs fois les paramètres à partir d'initialisations aléatoires différentes jusqu'à obtenir un résultat satisfaisant. Il faut faire également varier la taille de la carte ainsi que celle du voisinage. C'est un inconvénient de ce genre de méthode à paramètres. Il est éventuellement intéressant d'effectuer ces réglages de manière automatique, mais d'un autre coté, il paraît opportun d'avoir différents points de vue à analyser.

Il semble impossible voire inutile de garder tout le vocabulaire d'un corpus de données pour des traitements statistiques robustes. Il est clair que certains mots seront trop peu fréquents pour fournir une bonne estimation de leur probabilité ; ils sont

donc à supprimer. Les mots trop nombreux comme les articles ou mots de liaison n'apportent pas forcément beaucoup d'information. On se propose donc d'enlever d'un premier dictionnaire les mots qui sont soit trop représentés, soit pas assez pour le corpus. En pratique, nous classons l'ensemble du vocabulaire employé au total, par fréquence décroissante. Ensuite nous conservons environ les 1000 premiers plus fréquents en supprimant les quelques *trop* fréquents du début de la liste (mots outils principalement). De nombreuses méthodes alternatives de sélection de variables sont usitées en analyse textuelle. Parmi elles, on peut citer, dans le cas de textes labellés, une sélection des mots comme celle apportant le maximum d'information au sens de la théorie de l'information, et quantifié par l'indicateur d'information mutuelle. Dans le cas du classifieur naïf de Bayes, des auteurs ont également déterminé une taille empirique du vocabulaire à employer pour minimiser l'erreur ; ils ont obtenu justement un vocabulaire d'environ 1000 mots sur plusieurs corpus, mais cela n'empêche pas d'autres travaux récents d'utiliser avec succès un vocabulaire beaucoup plus large.

Pour les corpus servant aux tests, nous nous restreignons à environ mille composantes vectorielles, en conservant un vocabulaire supposé assez pertinent pour la discrimination effectuée par les algorithmes de carte auto-organisatrice.

Résumés INRIA Dans la publication[76] il est étudié un corpus d'environ 2000 résumés d'articles de l'INRIA. Les auteurs se servent essentiellement de l'AFC pour cette étude. Ces résumés sont issus des différents projets de l'institut que l'on peut regrouper en quelques domaines. Nous les projetons dans la suite.

Newsgroups WEB Les 20000 *news* cités dans [77] servent fréquemment de *benchmark*. Ils se répartissent dans une vingtaine de *newsgroups* comptant un millier de mails chacun. Les thématiques ont un large spectre : de l'informatique à la religion en passant par la santé et la philosophie. Nous les cartographions ci-après.

Monde Diplomatique Le corpus du Monde Diplomatique a l'intérêt d'être de taille importante, de l'ordre de 100000 documents, et d'embrasser un grand nombre de thématiques. Nous en construisons des cartes de distribution de mot.

4.2.1 Cas CASOM

4.2.1.1 Expériences sur les résumés INRIA

Nous projetons le corpus sur une carte topologique CASOM. On visualise alors la structure générale du corpus avec les indicateurs proposés. On peut vérifier sur cet exemple la bonne adéquation entre la projection factorielle des correspondances et la projection sur la carte auto-organisatrice à base de multinomiales. On présente avant tout la matrice U correspondant aux résumés en anglais, pour le tableau de mots affiché peu après :

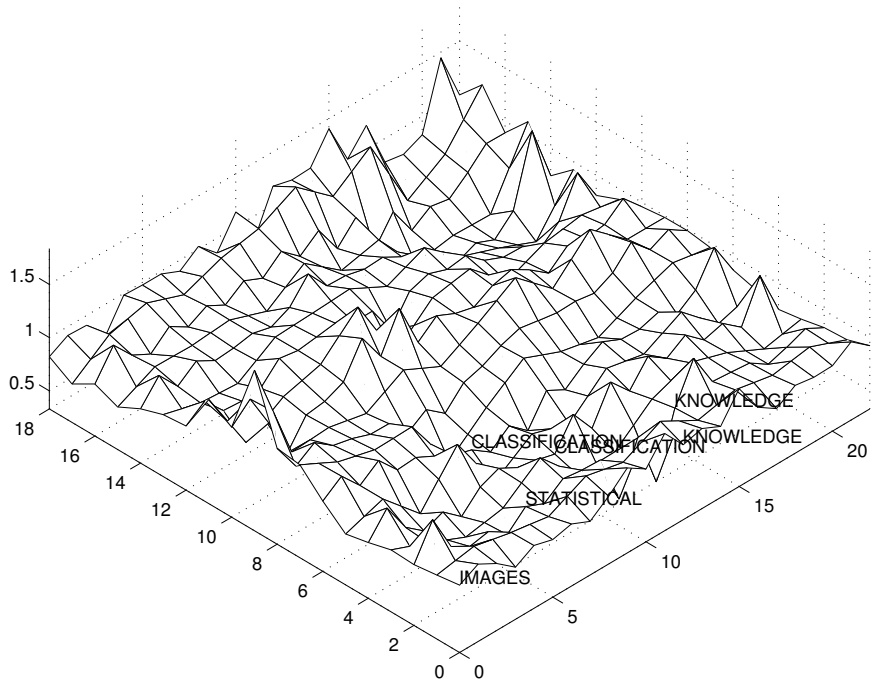


FIG. 4.12 – U-matrice des résumés, version anglaise

On affiche ci-après les projections factorielles pour un *early stopping* des versions de résumés en français, afin de vérifier empiriquement la propriété d'AFC non linéaire. Les projections montrent clairement une forme de grille rectangulaire déformée non linéairement, suivant une surface, avant déformation par spécialisation. On peut observer que les angles de la carte comptent le maximum d'inertie permettant d'affirmer que ce sont les thématiques les plus tranchées. Ensuite, il faudrait se reporter aux indicateurs d'inertie et de qualité pour dégager les zones révélées par la projection.

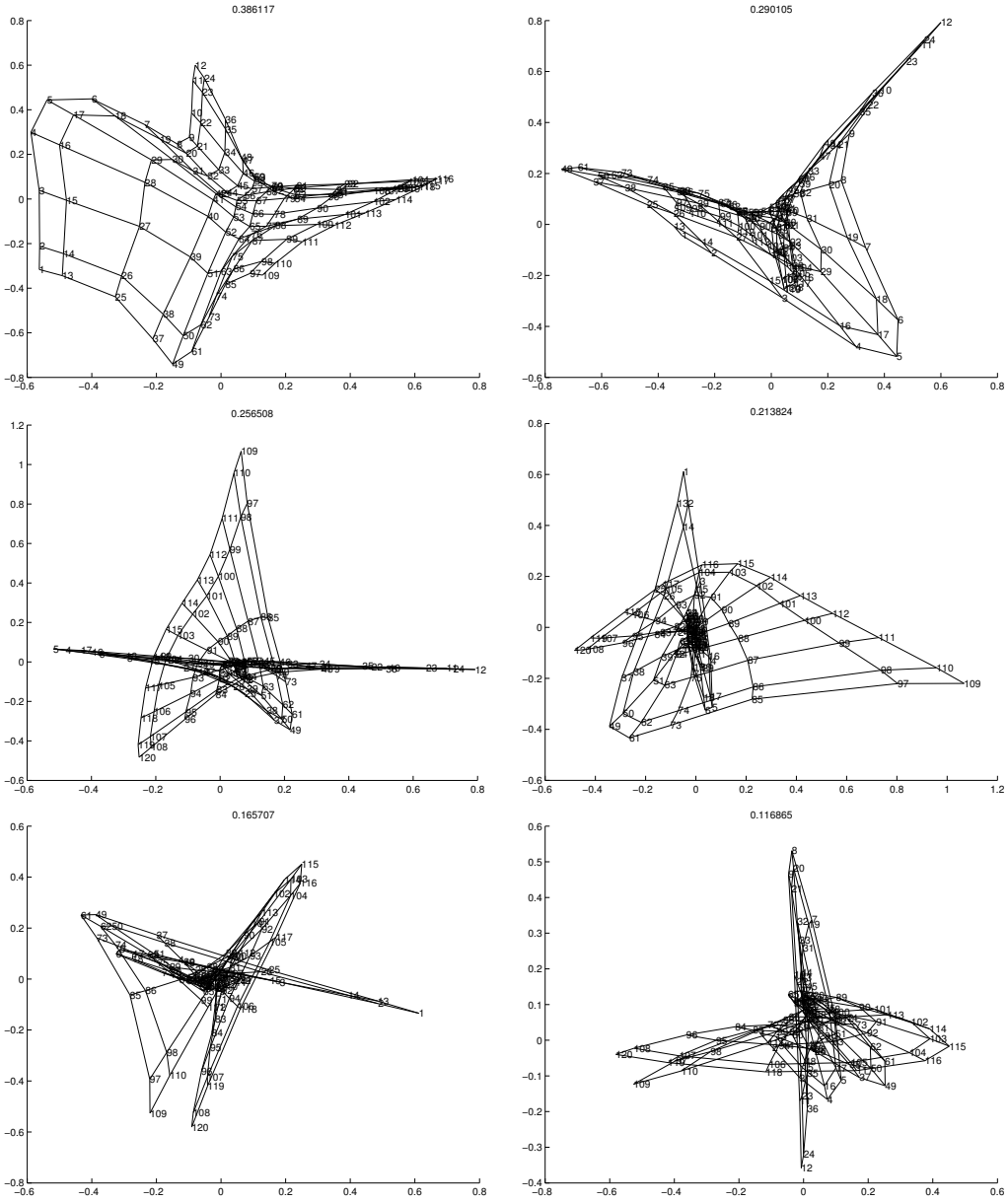


FIG. 4.13 – Plans factoriels (1,2) (2,3) (3,4) (4,5) (5,6) (6,7) et inertie projetée pour les multinomiales. Sur cet exemple, nous avons procédé à un arrêt avant convergence complète. Nous obtenons une forme montrant empiriquement le lien entre l'AFC et CASOM : une projection non-linéaire, résumant l'information contenue dans les plans de l'AFC des données. Les figures montrent également que le nuage projeté se rétracte et diminue rapidement en inertie expliquée.

Les relations de voisinage entre neurones sont bien conservées sur le graphique. On affiche les cartes d'activation pour la version en anglais à la suite du tableau de mots obtenu :

IMAGES	IMAGE	IMAGE	IMAGES	ROBOT	MODELS	KNOWLEDGE	THERE	MEMORY	DISTRIBUTED	DISTRIBUTED	CODE
IMAGE	MATCHING	IMAGES	SPACE	DYNAMIC	SIMULATION	CRITERIA	DIFFERENT	SIMULATION	MEMORY	APPLICATIONS	APPLICATIONS
SEGMENTATION	IMAGES	SEEN	DEVELOPED	SIGNAL	BOTH	CONTROL	DESIGN	SHARED	IMPLEMENTATION	PERFORMANCE	PERFORMANCE
POINTS	IMPORTANT	CLASSIFICATION	OBJECT	SPACE	CONTROL	DESIGN	KNOWLEDGE	PERFORMANCE	PROTOCOL	COMMUNICATION	QUERY
REAL	FEATURES	DEVELOPED	INFORMATION	MOTION	DYNAMIC	METHODS	CONTROL	PARALLEL	APPLICATIONS	NETWORKS	APPLICATION
OBJECTS	PARAMETERS	CONSTRAINTS	BOTH	VISION	SEEN	REPORT	DESIGN	ARCHITECTURES	SHARED	IMPLEMENTATION	IMPLEMENTATION
SURFACE	MODELS	PHASE	SEEN	DETECTION	MARKOV	CONCEPTS	USER	ARCHITECTURE	PERFORMANCE	APPLICATION	DISTRIBUTED
MOTION	REPORT	DIFFERENT	VISION	MODELING	REPORT	THEIR	MAIN	EXECUTION	SIMULATION	PROTOCOL	PROTOCOL
LEVEL	VISION	MODELS	TECHNIQUE	CRITERIA	MODELING	INTERFACE	ARCHITECTURE	PARALLEL	PROCESS	PROCESS	SOFTWARE
INFORMATION	TOOL	MORE	MORE	TECHNIQUE	ROBOT	EVALUATION	PRESENTS	IMPLEMENTATION	ARCHITECTURE	NETWORK	COMMUNICATION
DIFFERENT	GEOMETRIC	CONTROL	DISCRETE	CONSTRAINTS	DESCRIBE	TASKS	WORK	REPORT	ARCHITECTURES	MESSAGE	REPORT
LOCAL	PROPOSED	VISION	SEGMENTS	PHYSICAL	APPLICATION	THEY	CACHE	DESIGN	REPORT	INTERFACE	INTERFACE
PARAMETERS	ROBOT	TECHNIQUES	GOES	THOSE	MODELS	DEVELOPMENT	EXECUTION	PROCESSOR	LOCAL	LOCAL	COMPILER
SEQUENCE	METHODS	REPORT	APPLICATIONS	TOOL	RESEARCH	LANGUAGE	EVALUATION	MACHINE	PRESENTS	EACH	DESIGN
FEATURES	TECHNIQUES	TECHNIQUE	AVEC	ENVIRONMENT	MOTION	DOMAIN	SEEN	PROCESSING	MORE	PROPOSED	SUPPORT
IMAGES	OBJECT	ESTIMATION	STATISTICAL	DETECTION	SEEN	SEEN	KNOWLEDGE	PARALLEL	DISTRIBUTED	DISTRIBUTED	APPLICATION
CURVES	PARTICULAR	DIFFERENT	OTHER	HAND	THEY	KNOWLEDGE	EACH	MEMORY	PARALLEL	APPLICATIONS	APPLICATIONS
USEFUL	SHAPE	SPACE	ESTIMATION	MODELS	SIMULATION	PROBLEMS	THEY	DISTRIBUTED	MEMORY	SOFTWARE	LANGUAGE
IMAGE	STRUCTURES	TECHNIQUES	DIFFERENT	PHYSICAL	RESEARCH	MEMORY	INTO	SHARED	PROGRAMMING	COMMUNICATION	MESSAGE
SEEN	WELL	PROPERTIES	INFORMATION	OTHER	SIMULATIONS	STUDIED	LAME	ENVIRONMENT	APPLICATION	PARALLEL	DISTRIBUTED
RESOLUTION	ASSOCIATED	IMAGE	FUNCTION	LARGE	MANY	DESIGN	SEVERAL	MACHINES	DESIGN	PARALLEL	COMMUNICATION
POINTS	OTHER	IMAGES	FINALLY	METHODS	PREVIOUS	TASKS	PARTICULAR	PERFORMANCE	PROGRAMS	PROBLEM	PROTOCOL
DIFFERENT	REPORT	SEEN	APPROACHES	STRATEGY	STRATEGY	THROUGH	ALLOW	LARGE	EXECUTION	IMPLEMENTATION	IMPLEMENTATION
RECONSTRUCTION	INFORMATION	REPORT	SEEN	DESCRIBE	PROPOSE	ASPECTS	MEMORY	TECHNIQUES	COMMUNICATION	PROGRAMMING	DESIGN
CAMERA	LEVEL	STRUCTURE	RESEARCH	STRUCTURES	WHEN	BOTH	PERFORMANCE	IMPLEMENTATION	PROCESS	REPORT	SUPPORT
SYNTHETIC	MAIN	VISION	ONLY	ALLOWS	MULTIPLE	APPROACHES	SHARED	COMPUTER	FEATURES	ARCHITECTURE	ENVIRONMENT
REAL	WHEN	GIVEN	COMPARE	LINEAR	STRUCTURE	BECAUSE	ARCHITECTURE	PROCESSORS	SHARED	CONTROL	DESCRIPTION
SEQUENCES	INTO	THEIR	DETECTION	ALLOW	WORK	DEFINED	DIFFERENT	APPLICATION	PERFORMANCE	DEVELOPED	PARTIAL
EACH	CALLED	VARIOUS	IMAGES	MAIN	ACCURACY	PARTICULAR	BOTH	SEEN	ARCHITECTURE	PURPOSE	SEEN
GENERAL	DIFFERENT	PARAMETERS	MODELS	STATISTICAL	COMMON	SOLVING	PRESENTS	PROPOSED	APPLICATION	ARCHITECTURES	OBJECTS
IMAGES	MATRIX	LEVEL	MODELS	HAND	CLASSIFICATION	SEEN	EXECUTION	DISTRIBUTED	DISTRIBUTED	DISTRIBUTED	LANGUAGE
MOTION	GEOMETRY	POINT	DETECTION	OTHER	RELATIONS	THEY	PROCESSORS	PARALLEL	PARALLEL	IMPLEMENTATION	ENVIRONMENT
CAMERA	NUMBER	APPROACHES	TECHNIQUE	BOTH	VERY	VERY	EXECUTION	PROGRAMMING	PARALLEL	PROCESS	SOFTWARE
RECONSTRUCTION	IMAGES	VIEW	METHODS	STRUCTURES	THROUGH	IMPLEMENTATION	EACH	PROGRAMS	PERFORMANCE	REPORT	INFORMATION
IMAGE	METHODS	ESTIMATION	STRUCTURES	CLASSIFICATION	DYNAMIC	BECAUSE	PERFORMANCE	DETECTION	PROCESS	DYNAMIC	DESIGN
PARAMETERS	DIFFERENT	LINEAR	INFORMATION	LARGE	TREE	DESCRIBE	PARALLEL	PROTOCOL	COMPUTATION	DESIGN	DESCRIPTION
SCENE	LINEAR	REPORT	DISCUSS	STATISTICAL	MOST	IMPORTANT	GLOBAL	GLOBAL	SEQUENTIAL	PROGRAM	DEVELOPMENT
POINTS	STATISTICAL	ABLE	LINEAR	DESIGN	STRUCTURES	POINT	SEEN	CONTROL	PRESENTED	CODE	APPLICATIONS
REAL	COMPUTER	RANDOM	ASPECTS	OBJECTS	CONSTRUCTION	LOCAL	VERY	APPLICATIONS	MECHANISM	LANGUAGES	PROVIDES
MATRIX	OBJECTS	OTHER	MOST	EXACT	CONSIDERED	DESIGN	PROCESSOR	THEIR	LANGUAGE	FRAMEWORK	TOOLS
METHODS	IMPORTANT	SAME	GENERATION	RANDOM	FRAMEWORK	WORK	MULTIPLE	IMPLEMENTATION	PROGRAMMING	PROGRAMMING	PROGRAMMING
GEOMETRIC	CRITERION	SECOND	TOOLS	COMPLEXITY	PROPOSED	EVEN	EXPERIMENTS	DEGREE	COMMUNICATION	CONTROL	APPLICATION
SPACE	CONSTRAINTS	THEIR	OPTIMIZATION	VECTOR	INFINITE	NUMBER	LOCAL	FINITE	GENERATION	PROGRAMS	IMPLEMENTATION
ESTIMATION	OBJECT	TOOLS	STOCHASTIC	ACCORDING	SEGMENTS	COMPUTATION	SINCE	MEMORY	DISTRIBUTION	DESCRIBE	DOCUMENT
EXPERIMENTS	SPACE	VERY	BOTH	CONSIDER	CONTEXT	APPLICATIONS	DESIGNED	PRESENTED	MAIN	GENERATION	REPORT
POINTS	CONSTRAINTS	WITHIN	NUMBER	CLASSIFICATION	SPACE	WORK	PARALLEL	DISTRIBUTED	PROGRAMS	CONTROL	LANGUAGE
CAMERA	POINTS	IMAGE	SEQUENCE	EACH	NUMBER	BEEN	DISTRIBUTED	LOCAL	PARALLEL	LANGUAGE	SPECIFICATION
RECONSTRUCTION	POINT	PROGRAMS	LINEAR	ALLOWS	IMPORTANT	PARALLEL	MEMORY	PROCESS	PROGRAMS	EXECUTION	IMAGE
PARAMETERS	GENERIC	NUMBER	COMPLEXITY	CLASS	CONTEXT	NUMBER	PROCESSORS	THEY	POLICIES	GRAPH	VERIFICATION
EQUATIONS	GIVEN	EFFICIENT	SEQUENCES	MATCHING	VECTOR	IMPORTANT	EXECUTION	DISTRIBUTED	DISTRIBUTED	IMPLEMENTATION	ENVIRONMENT
STRUCTURE	BEEN	VERSION	ALLOWS	COMPLEXITY	ENVIRONMENT	EVEN	ONLY	PROGRAMS	PROPERTIES	REPORT	TOOLS
OTHER	ALGORITHM	LEVEL	THEIR	PROPOSE	SOLUTIONS	SEVERAL	WHEN	DETECTION	SEQUENTIAL	COMPONENTS	FORMAL
MOTION	NUMBER	SMALL	EACH	IMPORTANT	TECHNIQUE	EACH	THERE	PRESENTED	DIFFERENT	ARCHITECTURE	SYNCHRONOUS
SCENE	WHEN	PROCESSING	PROPOSE	EFFICIENT	MOST	DESIGN	TASK	GLOBAL	REPORT	SOFTWARE	PROGRAMMING
PROPOSED	DESCRIBE	DOMAIN	WHEN	DIFFERENT	SEEN	PROCESSING	EACH	APPLICATIONS	INTO	BUILD	DESCRIPTION
SETS	THREE	COMPUTER	WELL	SECOND	ONLY	PROPOSE	VERY	GRAPH	PROVE	DEFINED	PROGRAM
POINT	CITE	ALLOWS	IMAGE	SPACE	VARIABLES	OTHER	PROPOSED	COMPUTATION	DYNAMIC	DYNAMIC	IMPLEMENTATION
ONLY	SOLUTIONS	MOST	COMPLEX	ACCORDING	PROPOSE	ARCHITECTURES	THREE	APPLICATIONS	CLASS	CLASS	TECHNIQUES
GIVEN	PREVIOUS	STRUCTURE	SECOND	CLASSES	COMPUTED	BOUNDED	WHOLE	SEEN	ENVIRONMENT	ENVIRONMENT	GENERATION
IMAGE	TECHNIQUES	COMPLEX	BOTH	FORMAL	OTHER	COMMUNICATION	SOLUTIONS	MORE	GIVEN	PROGRAM	UNIT
POINTS	POINTS	SPACE	METHODS	NUMBER	NUMBER	MEMORY	DISTRIBUTED	DISTRIBUTED	DISTRIBUTED	VERIFICATION	LANGUAGE
CONVEX	SIMPLE	ROBOT	DEGREE	IMPLEMENTATION	THAN	NUMBER	NUMBER	EXECUTION	PROPERTIES	LANGUAGE	PROGRAMMING
LINE	GIVEN	POINT	EXACT	CLASSIFICATION	ONLY	PROCESSORS	MESSAGES	PROGRAM	CONTROL	MECHANISM	SPECIFICATION
GIVEN	POINT	RESOLUTION	COMPUTATION	APPLICATION	OPTIMAL	SCHEDULING	SIZE	GLOBAL	GRAPH	PROGRAMMING	LANGUAGES
OBJECTS	SPACE	REPRESENTATION	SEGMENTS	VALUES	VARIABLES	SEEN	STRUCTURE	COMPUTATION	PROGRAM	STRUCTURES	ABSTRACT
POINT	WHERE	LINEAR	IMPLEMENTATION	VALUES	EACH	GENERAL	EXECUTION	STATE	LEVEL	PARALLEL	OBJECT
PROBLEMS	ROBOT	MOST	MOST	PROPOSE	MOST	GIVEN	ONLY	GRAPH	COMPUTATION	STATE	HUMAN
PLANE	EFFICIENT	EFFICIENT	EACH	CONSISTS	CLASSIFICATION	ONLY	POSSIBLE	GENERAL	EACH	VARIABLES	SEMANTICS
INITIAL	EXACT	GENERAL	MULTIPLE	COMPLEXITY	SEVERAL	POSSIBLE	PROGRAMS	MESSAGES	DETECTION	EXECUTION	STRUCTURES
NUMBER	STRATEGY	FUNCTIONAL	TECHNIQUE	VALUE	SOFTWARE	WHEN	OVER	PROPERTY	OTHER	INPUT	APPLICATIONS
SEGMENTS	COMPUTE	METHODS	SOLVING	EFFICIENT	LARGE	VERY	VECTOR	GIVEN	OPERATORS	PROGRAM	FEATURES
QUERY	LINEAR	WHERE	EXPLICIT	GENERAL	STRUCTURE	VECTOR	EFFICIENT	SEQUENCES	PROCESSES	PROGRAMS	REPORT
APPLY	OPTIMAL	GEOMETRIC	PLANE	SIMULATIONS	GOOD	SPACE	BEING	CONTEXT	OVER	SOURCE	TOOL
STRUCTURE	SOLUTIONS	IMAGE	BEING	VERY	CITE	OPTIMAL	CONSTRUCTION	WHEN	CALLED	TOOL	FORMALISM
COMPUTE	EACH	SEEN	MODELS	DISCUSS	MEMORY	PARALLEL	UNDER	PROPERTIES	DEFINED	INTO	SOURCE
SURFACE	OPTIMAL	REPRESENTATION	POLYNOMIAL	CODE	NUMBER	SCHEDULING	SCHEDULING	SEQUENTIAL	CONSISTENCY	GRAPH	SEMANTICS
MESH	STRATEGY	DIFFERENT	MORE	LINEAR	COMPLEXITY	COMMUNICATION	INTO	CONSISTENCY	GRAPH	FORMALISM	LANGUAGE
PART	GIVEN	POLYNOMIAL	THEIR	CODE	PARALLEL	TASKS	CONSISTENCY	STATE	OBJECT	OBJECT	PROGRAMMING
EXAMPLES	FINITE	MORE	SOLUTIONS	ONLY	MOST	PARALLEL	EXECUTION	CLASS	PROGRAMMING	LANGUAGES	LANGUAGES
PROPOSED	CONSIDER	COMPUTE	ALGORITHM	THEIR	WHERE	PROCESSOR	PROCESSORS	TASKS	SEQUENTIAL	EACH	PROGRAM
APPLICATION	CRITERIA	DETECTION	CLASSICAL	EACH	PROBLEMS	COMPLEXITY	CONSIDER	CALLED	ONLY	MORE	ABSTRACT
SURFACES	CLASS	APPROACHES	PROPOSE	GEOMETRIC	POLYNOMIAL	TASK	GENERAL	CONTEXT	CRITERIA	CLASS	LOGIC
GIVEN	PROGRAM	PARTICULAR	DEGREE	GIVEN	CONSIDER	WHERE	WHERE	STRUCTURE	EACH	LANGUAGES	FUNCTIONAL
MESHES	PROGRAMS	VARIOUS	STRUCTURE	THAN	OBTAIN	WHEN	CONSTRAINTS	OPERATIONS	THEORY	PROGRAM	NATURAL
REPORT	REPORT	POINT	ALLOWS	ALLOWS	WELL	PROBLEMS	TIMES	REPRESENT	SEEN	DIFFERENT	DYNAMIC
GENERATION	POINTS	RESOLUTION	THEORY	CONSISTS	LINE	FIND	FILED	DOES	MATCHING	DESCRIBE	IMPLEMENTATION
SEVERAL	PART	GIVEN	MOST	FUNCTION	OPTIMAL	LINE	TASK	PROCESSES	QUERY	DESCRIBED	EACH
SHAPE	APPLICATION	PROBLEMS	VARIABLES	HERE	EACH	GENERAL	OPTIMAL	DISTRIBUTED	FORMALISM	LANGUAGE	STRUCTURES
GEOMETRIC	NUMERICAL	SOLUTIONS	IMPLEMENTATION	BOTH	PROCESSOR	POSSIBLE	DISTRIBUTED	GENERAL	IMPLEMENTATION	IMPLEMENTATION	PROGRAMS
NUMERICAL	COMPUTE	THEORY	INITIAL	SMALL	THAN	EACH	WHEN	PROGRAMMING	CLASS	UNIT	FORMALISM
FLOW	GIVEN	MATRIX	MATRIX	COMPLEXITY	ONLY	STRUCTURE	GRAPH	GRAPH	CONTROL	STRUCTURES	SEMANTICS
SHAPE	APPLICATIONS	MARKOV	MATRIX	PROBLEMS	USE	OPTIMAL	SCHEDULING	STATE	RULES	PROPERTIES	PROGRAM
FIELD	NUMERICAL	ALLOWS	POLYNOMIAL	MATRIX	NUMBER	GENERAL	STRUCTURE	CLASS	STRUCTURE	THEIR	LANGUAGE
NUMERICAL	METHODS	DESIRE	PROBLEMS	POLYNOMIAL	POLYNOMIAL	SCHEDULING	GRAPHS	EACH	THEORY	NOTION	PROOF
MESH	SIMULATION	METHODS	COMPUTING	GIVEN	TIMES	GRAPH	NUMBER	OBJECTS	CALLED	CONTEXT	PROPERTIES
MESHES	PROBLEMS	FUNCTIONS	PART	MATRICES	LINES	PROCESSOR	TREES	PROCEDURE	BUILD	ALGEBRAIC	LANGUAGES
FLOW	REPORT	APPROXIMATION	METHODS	THEORY	LINEAR	DRIVE	TASKS	GLOBAL	OPTIMAL	SEQUENCES	PROGRAMMING
SIMULATION	ALLOWS	TERMS	ALGEBRAIC	FORM	GRAPH	POSSIBLE	PARTICULAR	CALLED	GRAPH	FUNCTIONAL	LOGICAL
EQUATIONS	SIZE	POINT	APPLICATIONS	ASYMPTOTIC	NETWORK	DISTRIBUTION	CONSTRUCTION	DISTRIBUTED	BUILD	GIVEN	TREES
COMPUTATION	APPLIED	INTO	NUMERICAL	DEVELOPED	THEIR	MODELS	GENERATED	TREES	STRUCTURES	FUNCTIONS	LOGIC
EQUATION	EFFICIENT	PACE	COMPUTATION	GENERAL	GOOD	PROPOSE	DISTRIBUTED	GRAPH	CLASSES	DEFINED	FRAMEWORK
PROBLEMS	EXAMPLES	WHERE	GIVEN	RESULT	SIMPLE	CLASS	PARALLEL	INFORMATION	TERMS	FUNCTION	SPECIFICATIONS
SOLUTION	INITIAL	CONSTRAINT	LARGE	UNDER	SPACE	PARAMETERS	EACH	EVENT	GENERAL	USES	PROVE
CITE	SOLVE	PROPOSED	NUMBER	ALGEBRA	COMPLEXITY	RESPECT	TASK	SEMANTICS	EXTENSION	EITHER	TYPE
FORMULATION	ILLUSTRATE	REPORT	TECHNIQUE	POINTS	COMPUTATION	SOLUTIONS	ALLOWS	LINEAR	CURRENT	PROGRAM	PROGRAMS
NUMERICAL	NUMERICAL	APPROXIMATION	FUNCTIONS	PART	MODELS	STOCHASTIC	TIMES	CONSTRUCTION	PROPERTIES	THEIR	PROOF
EQUATIONS	METHODS	SOLUTION	METHODS	METHODS	ASYMPTOTIC	SIMULATION	TREES	TREE	OTHER	FUNCTIONS	LOGIC
FLOW	TERMS	METHODS	LINEAR	PROBLEMS	NETWORKS	NETWORKS	DISTRIBUTED	TREES	THOSE	PROCESSES	CALCULUS
SOLUTION	WHEN	LINEAR	MODELS	DEVELOPED	COMPUTATION	POINT	OPTIMAL	DISTRIBUTED	DEFINE	LANGUAGE	LANGUAGE
DIFFERENT	CONSIDER	MATRICES	MATRICES	ASYMPTOTIC	NUMBER	PROCESSES	PROCESS	PROCESSES	CONSISTENCY	FORM	TERMS
FLOWS	SOLUTION	LARGE	STRUCTURES	FUNCTIONS	NUMBER	NUMBER	PROCESSORS	PROVE	THEORY	PROVE	THEORY
SIMULATION	FLOW	FIELD	WORK	WHERE	STOCHASTIC	AVAILABLE	NUMBER	RELATIONS	OBJECTS	BEHAVIOUR	PROGRAM
CHEMES	SIMULATION	ILLUSTRATE	FUNCTION	REPORT	STRUCTURES	COMMUNICATION	EACH	EXPLICIT	WHERE	SPECIAL	PROPERTIES
FINITE	NUMBER	WHEN	CONVERGENCE	TIMES	REPRESENT	DISTRIBUTED	PROCESSOR	LIKE	THEIR	THEREFORE	PROVE
PRESENTED	CONVERGENCE	TERMS	MANY	THEORY	PROCESS	WHEN	TASKS	MEASURES	CLASS	PROPERTIES	RULES
REPORT	ILLUSTRATE	PROPOSED	EFFICIENT	WHEN	RELATED	PROVIDE	POLICIES	NOTION	CALLED	NOTION	GENERAL
APPLICATION	DECOMPOSITION	TECHNIQUES	STATE	GENERAL	SYMBOLIC	BOTH	OPTIMAL	CONSTRUCTION	OPTIMAL	NOTION	PROGRAMS
APPROXIMATION	MORE	COMPUTING	MATRIX	ONLY	POINT	CHARACTERISTICS	CONSIDER	PROCESS	GRAPH	PROVE	CONSISTENCY
PROPOSED	THEIR	FINALLY	PROBLEMS	CASES	COMPUTATIONAL	NEEDED	OBJECTS	SIMPLE	MAIN	PARTICULAR	NOTION
BOUNDARY	OBTAIN	USES	LARGE	GENERAL	RESEARCH	COMPLETE	MEANS	SIMPLE	SEEN	DEFINE	TERM
EQUATIONS	PROBLEMS	FUNCTION	FUNCTION	MEASURES	MODELS	NETWORKS	GRAPH	TRANSITION	OPTIMAL	NOTION	LANGUAGE
NUMERICAL	METHODS	CONSTRAINTS	ASYMPTOTIC	MODELS	THEORY	BOUNDS	PROCESSES	SETS	PARTICULAR	PROCESSES	EXTENSION
METHODS	LINEAR	METHODS	COMPUTE	ASYMPTOTIC	PROCESS	RANDOM	PROVE	INFINITE	CLASS	LOGIC	TYPE
SOLUTION	FIELD	CONDITIONS	WHEN	STATE	PARAMETERS	TIMES	STRUCTURE	GRAPH	GRAPH	GIVE	CALCULUS
MESH	EQUATIONS	CONVERGENCE	GIVE	FUNCTION	GRAPH	NUMBER	NUMBER	FINITE	OTHER	VALUES	TYPES
CHEME	SOLUTION	FUNCTIONS	FUNCTIONS	FUNCTIONS	RANDOM	WHEN	SIZE	DEFINED	NETS	CONSIDER	TERM
REPORT	CONVERGENCE	LINEAR	PROPOSED	METHODS	NUMBER	GRAPH	GIVEN	COMPUTATIONS	SIMPLE	EXTENSION	LAMBDA
FLOW	SECOND	WHEN	LINEAR	CLASS	DISTRIBUTION	WHERE	SENSE	GRAPH	THEIR	CALCULUS	TERMS
FINITE	TECHNIQUES	THEORY	NONLINEAR	PROCESSES	GENERAL	COST	FINITE	TREE	THOSE	PARTICULAR	REDUCTION
PRESENTED	OPTIMIZATION	NUMBER	DIFFERENTIAL	LARGE	MARKOV	POINT	OBTAINED	GENERAL	REDUCTION	INTO	FUNCTIONAL
DOMAIN	VERY	PROBLEMS	NUMERICAL	INFORMATION	LARGE	AVERAGE	DESIRE	LINEAR	BUILD	PROPERTIES	SPACE
SOLUTIONS	NUMERICAL	SOLUTIONS	APPLICATION	LINEAR	FRAMEWORK	PROVIDE	EFFICIENT	PROPERTIES	WELL	EXPLICIT	CALCULUS
EQUATION	EQUATION	UNDER	TECHNIQUES	FINITE	PROBLEMS	COMMUNICATION	UNDER	SAME	CONVEX	POWER	NOTION
FLOW	REPORT	GIVE	CONSIDER	ESTIMATION	SEE	THERE	SEEN	CRITERION	BUILD	INTRODUCED	MORE
PROPOSED	OBTAINED	VALUE	DYNAMIC	DISCRETE	TECHNIQUES	SMALL	COMPLETE	STRUCTURE	HIGHER	UNDER	PROGRAMMING
EQUATIONS	SOLUTION	CONTROL	CONTROL	STATE	RANDOM	TIMES	TREE	PROCESS	SETS	WEAK	CACHE
EQUATION	PROBLEMS	OPTIMAL	FUNCTIONS	CONDITIONS	PROCESS	SERVICE	TIMES	PETRI	POINT	TERMS	LAMBDA
NUMERICAL	EQUATIONS	FUNCTION	FUNCTION	MARKOV	QUEUE	TRANSITION	CLASS	SEARCH	TERM	CALCULUS	
FINITE	EQUATION	PROBLEMS	CONTINUOUS	PROCESS	NETWORK	NETWORKS	NUMBER	CLASS	PARTICULAR	PROVE	TYPE
CONDITIONS	CONDITIONS	NONLINEAR	CONDITIONS	UNDER	FINITE	WHEN	WHERE	GRAPH	NUMBER	CLASSICAL	TERMS
CHEME	CONVERGENCE	SOLUTIONS	NETWORK	CONVERGENCE	RATE	NUMBER	PETRI	SEARCH	SETS	GIVEN	CALCULUS
BOUNDARY	NUMERICAL	SOLUTION	PROPOSED	FINITE	PROBABILITY	STOCHASTIC	RANDOM	FINITE	CITE	SCHEMES	REDUCTION
ELEMENTS	LINEAR	EQUATION	CONVERGENCE	CONSIDER	SERIES	RANDOM	GIVEN	GRAPH	SETS	TYPE	TYPES
SOLUTION	SOLUTIONS	CONVERGENCE	NONLINEAR	STABILITY	UNDER	PROCESS	DISTRIBUTED	GIVEN	GIVEN	PROPERTIES	THEORY
METHODS	EXISTENCE	SOLUTIONS	FINITE	CONTROL	CONSIDER	PROCESSES	NETS	TREE	INTRODUCE	GIVE	EXPLICIT
CHEMES	TYPE	CONSTRAINTS	STATE	WHEN	GENERAL	DISTRIBUTION	PROCESS	PROBLEMS	GIVE	HIGHER	STRONG
PRESENTED	PROVE	CONDITION	ASYMPTOTIC	GENERAL	SERVICE	VARIABLES	POLICIES	POLYNOMIAL	LAMBDA	INTRODUCE	TERM
SECOND	ERROR	GIVE	APPROXIMATION	COST	PROCESSES	POLICIES	KNOWN	STEP	TYPE	THEIR	WHERE
LINEAR	GIVE	APPROXIMATION	LINEAR	SPACE	STABILITY	WHERE	NETWORKS	WHERE	EXTENSION	FINALLY	SIZE
PROBLEMS	OPTIMAL	STATE	METHODS	NETWORK	GIVEN	EVENT	CONSIDER	GENERAL	SHOWN	USES	PROPERTIES

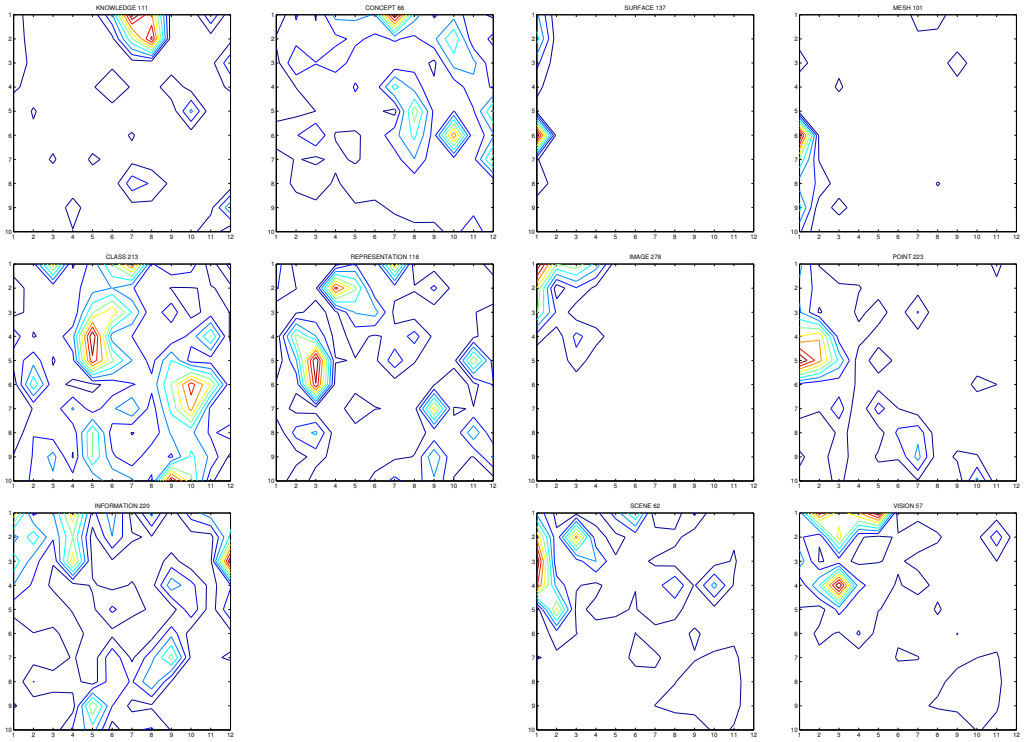


FIG. 4.14 – Nous illustrons la propriété de recherche d'information du modèle. Plus la couleur est rouge, et plus la probabilité est élevée. En haut de chaque carte figure le mot (ou premier du groupe) de vocabulaire correspondant. Cela correspond à des cartes d'activation avec des lignes de niveau pour la somme des probabilités des multinomiales pour les questions "*knowledge*", "*concept concepts*" et "*class classes classification*". N'importe quel indicateur pourrait être utilisé au lieu des probabilités, mais la propriété des valeurs bornées pourrait se perdre. Les figures montrent comment la carte auto-organisatrice se comporte : elle est activée sur différentes zones selon les diverses thématiques du corpus. Par exemple, le terme *knowledge* est statistiquement proche du terme *concept*, comme en témoigne les courbes superposables obtenues à la suite de la question. Ces figures montrent les cartes d'activation pour la thématique *visual* : (*image, point, surface, mesh, scene, vision*). On visualise directement les fortes liaisons statistiques de certains mots du vocabulaire qui apparaissent de façon très probables dans les mêmes zones cartographiques.

4.2.1.2 Expériences sur les articles de *newsgroups*

L'ensemble des *news* a été projeté sur une carte de taille 5*4, valant le nombre de groupes différents de *news*. On reconnaît pratiquement l'ensemble des groupes mais certaines thématiques sont peu séparées et se partagent plusieurs multinomiales. Dans un optique d'extraction de connaissances par visualisation, nous pouvons observer rapidement l'organisation générale du corpus, de ses diverses thématiques et de leurs relations.

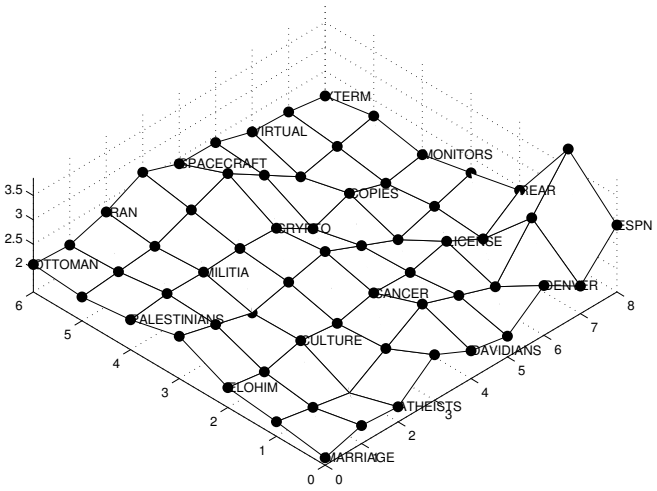


FIG. 4.15 – U-matrice des *news* pour la grille ci-dessous

MARRIAGE	ATHEISTS	DAVIDIANS	DENVER	ESPN
APARTMENT	SUBJECTIVE	CULT	SEMI	STATS
GRACE	GODS	TEAR	JOKE	PLAYOFFS
MATTHEW	ASSUMPTION	TEXAS	BUSH	BRAVES
SUMGAIT	JUDGEMENT	ASSAULT	HOSPITAL	PITCHER
LUKE	LOGICAL	SUICIDE	ROCK	CHICAGO
VERSE	PERSPECTIVE	RAID	TRAINING	DETROIT
ETERNAL	ASSERTION	TRIAL	BRIAN	PITCHING
TESTAMENT	BELIEVED	FEDS	HITTING	PITTSBURGH
REVELATION	TOPIC	KNOCK	SHOTS	MORRIS
ELOHIM	CULTURE	CANCER	LICENSE	REAR
MORMON	MALE	MEDICINE	OWNERS	RIDING
UNTO	EXAMPLES	DIET	SALES	FORD
GODS	MINORITY	PATIENT	DETECTOR	TIRES
DOCTRINE	DECISIONS	BRAIN	PASSED	WIRING
MOVEMENT	HOMOSEXUALS	CANDIDA	TAXES	BIKES
BIBLICAL	IGNORANT	SYMPTOMS	OWNER	BRAKE
LEADERS	SCHOOLS	CHRONIC	WAITING	HELMET
JOSEPH	THREAT	DOCTORS	ACCIDENT	NEUTRAL
VERSES	RATIONAL	LEVELS	RIDING	HONDA
PALESTINIANS	MILITIA	CRYPTO	COPIES	MONITORS
PALESTINIAN	BEAR	ENCRYPTED	ENGINES	CHANNEL
ISLAMIC	CONSTITUTIONAL	PHONES	LETTERS	ELECTRONICS
LEBANON	SUPREME	COMMUNICATIONS	STUDENT	VOLTAGE
SERBS	WEAPON	PROPOSAL	RECORDS	SONY
CIVILIANS	NUCLEAR	SCHEME	SALES	LASER
BOSNIA	LIBERTARIAN	AGENCIES	COMPONENTS	STEREO
JERUSALEM	ORGANIZED	AGENCY	STORE	PORTABLE
NAZIS	FIGHTING	CRYPTOGRAPHY	ANALYSIS	ITEMS
NATIONS	PAPERS	EXPORT	CORP	WARRANTY
OTTOMAN	IRAN	SPACECRAFT	VIRTUAL	XTERM
GREECE	JOIN	LUNAR	PLANE	BIOS
AZERI	JUNE	MARS	OBJECTS	POSTSCRIPT
EMPIRE	REGION	VENUS	CRYPTOGRAPHY	SIMMS
PARAGRAPH	RUSSIA	LARSON	FUNCTIONS	OPENWINDOWS
REPUBLIC	TROOPS	SATELLITE	DISTANCE	CACHE
KARABAKH	GREECE	PROBE	TELEPHONE	BYTE
MASSACRE	NUCLEAR	ASTRONOMY	PROCESSING	SHELL
KURDS	TRAINING	TEMPERATURE	SESSION	PROCESSOR
VILLAGE	POTENTIAL	PROPULSION	FREQUENCY	TRANSFER

4.2.1.3 Expériences sur les résumés du Monde diplomatique

Nous avons voulu essayer la méthode sur un corpus plus vaste de l'ordre de la centaine de documents. La base projetée consiste en I=98432 documents pour une taille de vocabulaire retenu de J=964. Les textes sont des paragraphes d'articles du monde diplomatique pendant une dizaine d'années. On remarque que les cartes construites de taille 7*7 sont très petites au regard de la taille du corpus, mais déjà se dégage des résultats non dénués d'un certain sens dont la complète pertinence serait intéressante à quantifier pleinement. Il faudrait par exemple chercher à évaluer la qualité des corrélations établies ici, et ce sur différentes tailles de carte croissantes. On a construit les cartes suivantes.

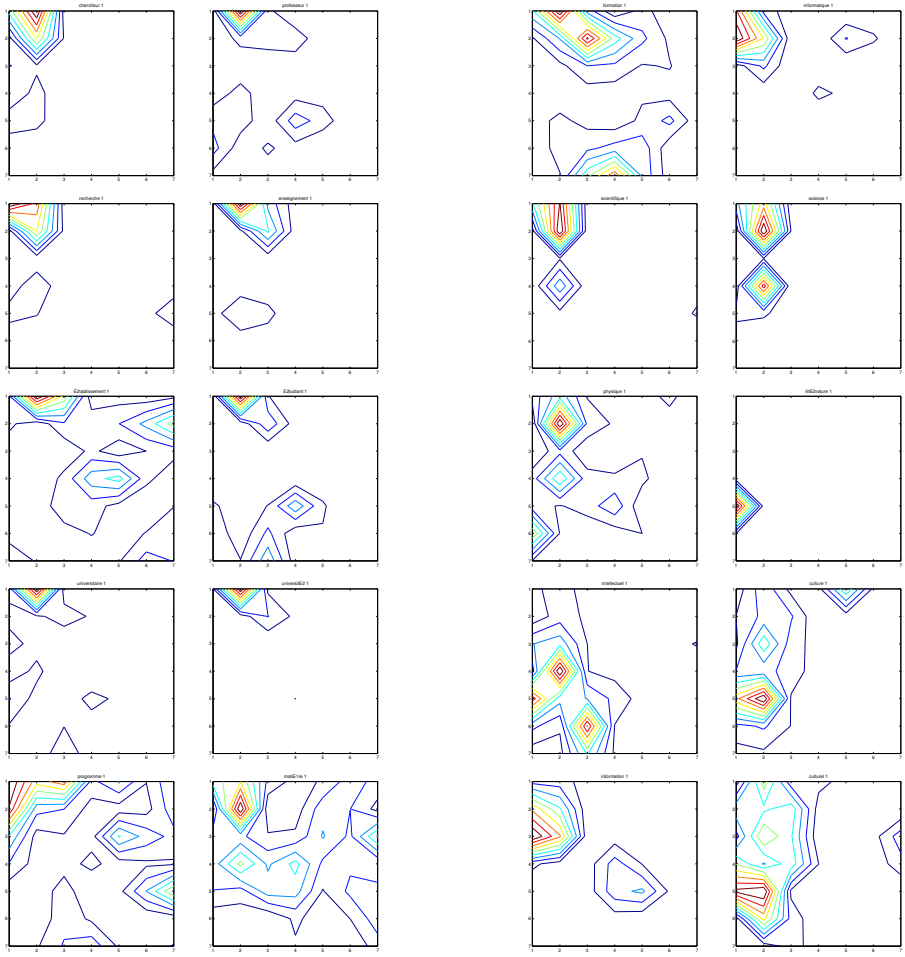


FIG. 4.16 – Carte sur le thème de l'enseignement dans le monde diplomatique. En observant les modes (couleur rouge), on visualise des liens entre certains mots du vocabulaire.

4.2.2 Cas TNEM

Pour cet exemple, nous initialisons grossièrement le réseau par quelques pas de la méthode CASOM. Nous allons observer principalement l'utilité du paramètre de lissage β puisqu'il n'est pas présent dans le SOM. Nous montrons empiriquement sur un corpus d'exemples (2000 résumés INRIA, 500 premiers mots du vocabulaire) son influence dans l'auto-organisation de la carte sur la figure ci-dessous :

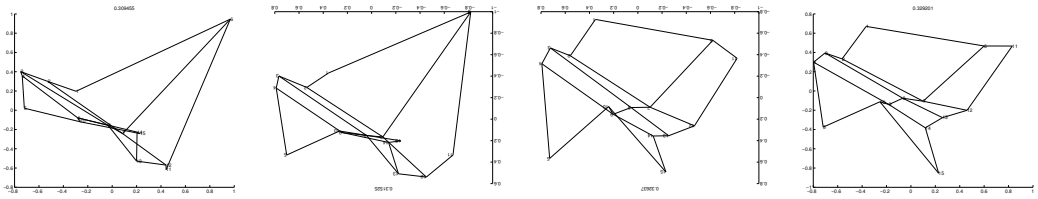


FIG. 4.17 – Premier plan factoriel des multinomiales obtenues, pour de gauche à droite : $\beta = 1$, 5, 10, 12.5, et 15. Pour comparer les vues nous avons effectué des rotations puisque les plans factoriels sont recalculés à chaque fois.

En prenant garde aux mauvaises interprétations de la projection, on peut observer les bords de la grille sur le premier plan de l'AFC des multinomiales. On voit la carte se déplier plus ou moins suivant la valeur du paramètre pour une initialisation identique. Il serait intéressant de le calculer de façon automatique afin que la carte soit la plus dépliée possible. On obtient :

LOGIQUE	PROTOCOLE	PERFORMANCES	PROCESSEURS	CODES	FINIS
LAMBDA	COMMUNICATION	CACHES	CODE	CODE	ELEMENTS
TERMES	PARALLELES	CACHE	INSTRUCTIONS	LINEAIRES	MAILLAGES
	MESSAGES	EVALUATION	PARALLELE	BOUCLES	SCHEMAS
SUPERIEUR	PROTOCOLES	PARTAGEE	PROCESSEUR	OPTIMISATION	NUMERIQUES
REECRITURE	COMPILATEUR	ARCHITECTURE	MACHINE	DIMENSION	NUMERIQUE
SUBSTITUTIONS	PERFORMANCES	PERFORMANCE	AUTOMATIQUE	STRATEGIE	MAILLAGE
NORMALISATION	PARALLELISME	ARCHITECTURES	BOUCLES	NUMERIQUES	SCHEMA
REDUCTION	COHERENCE	MACHINES	OPTIMISATION	EFFICACITE	ECOULEMENTS
TYPAGE	ENVIRONNEMENT	SIMULATION	GENERATION	GEOMETRIE	EQUATION
SOURCES	CONCEPTION	TRAFIC	PROGRAMME	LOGICIEL	APPROXIMATION
REQUETE	ARCHITECTURES	PARALLELES	ORDONNANCEMENT	SIMULATION	VOLUMES
UNIFICATION	CODE	ACCES	ARCHITECTURE	NUMERIQUE	SIMULATION
EXPLICITES	MECANISMES	PROCESSEURS	PERFORMANCES	OPTIMAL	NAVIER-STOKES
LANGAGES	MACHINES	INSTRUCTIONS	IMPLEMENTATION	PERFORMANCES	ONDES
LANGAGES	CRITERES	ORDONNANCEMENT	CONSTRUCTION	MESURES	EQUATION
SEMANTIQUE	EVALUATION	TACHES	ARBRES	IDENTIFICATION	CONVERGENCE
PREUVE	DETECTION	PARALLELES	COMMANDE	ASYMPTOTIQUE	APPROXIMATION
PIS	REPARTIS	COMMUNICATION	DIFFUSION	LINEAIRES	OPTIMAL
LOGIQUE	REPARTI	PETRI	EFFICACE	MESURE	OPTIMISATION
SPECIFICATION	INTERFACE	GRAPHES	EXEMPLES	CONSIDERE	ONDES
CONCEPTION	NIVEAU	GRAPHE	ASPECTS	CONVERGENCE	CONDITION
VERIFICATION	CONCEPTION	CONSIDERONS	MECANISMES	VARIABLES	QUADRATIQUE
DESCRIPTION	ETATS	COMMUNICATIONS	CLASSES	FILTRAGE	ERREUR
REGLES	UTILISATEUR	BORNES	ERREUR	LIMITES	NUMERIQUE
OUTILS	NOUVEAUX	OPTIMAL	DECRIVONS	LOIS	NUMERIQUES
DEVELOPPEMENT	RESSOURCES	COMPLEXITE	PERMETTANT	OPTIMISATION	ESTIMATION
DOCUMENT	ARCHITECTURE	EVALUATION	CONTEXTE	NUMERIQUE	DECOMPOSITION
GENERATION	DEVELOPPE	PROCESSEURS	TRAVAUX	TERME	CONVEXE
PREUVES	MESSAGES	PAPIER	PROPOSEE	BASEE	LINEAIRES
ENVIRONNEMENT	NOTION	ASYMPTOTIQUE	ARBRES	VARIABLES	CHAMP
SPECIFICATION	PETRI	ASYMPTOTIQUES	COMPLEXITE	VALEURS	CONVERGENCE
OUTILS	PROCEDURE	SYNTHESE	ARBRE	STABILITE	OPTIMISATION
SEMANTIQUE	CONDITION	LINEAIRES	VARIABLES	EXPRESSION	CRITERE
VERIFICATION	SYNTHESE	MOYENNE	MOYENNE	BORNE	OPTIMALE
CONCEPTION	CORPS	ELEMENTAIRES	CONSTRUCTION	VALEUR	APPROXIMATION
SIGNAL	REECRITURE	AUTOMATIQUE	LINEAIRES	EFFICACE	EQUATION
SPECIFICATIONS	EXISTENCE	FORMEL	DELAUNAY	COUT	ESTIMATION
DESCRIPTION	NECESSITE	TERMES	LOIS	RESULTAT	EXISTENCE
INTERFACE	TESTS	CLASSES	MESURES	EVALUER	COMMANDE
CONNAISSANCES	REGLES	ALGEBRE	STABILITE	COMPLEXITE	CONDITION
PREUVE	FORMES	PETRI	PRATIQUE	ESTIMATION	COURBURE
REPRESENTATION	RESULTAT	GRAPHE	STATIONNAIRE	CALCULER	ASYMPTOTIQUE
INFORMATION	DETERMINER	SERIE	CONSIDERONS	PROPRIETE	INFINI
FORMELLE	EQUIVALENCE	COMPLEXITE	UNIFICATION	TAUX	LONG
ATTENTE	GRAPHE	MATRICES	TACHES	ROBOT	IMAGE
SERVICE	SIGNAL	THEOREME	ORDONNANCEMENT	MATRICE	MOUVEMENT
FILES	EVENEMENTS	POLYNOMES	RELATIONS	TACHES	CONTOURS
FILE	GRAPHES	NOMBRES	VARIABLES	OBJET	SEGMENTATION
GRAPHES	REPRESENTATION	RECHERCHE	COMMANDE	SCENE	SURFACE
SERVEUR	AUTOMATES	DYNAMIQUE	COMPLEXITE	MOUVEMENT	RECONSTRUCTION
GRAMMAIRES	ORDRES	ARBRES	MATRICES	COMPLEXITE	ESTIMATION
CLIENTS	TRAITEMENT	GRAPHES	REEL	RECONSTRUCTION	REELLES
DISTRIBUTION	SEQUENCES	OPERATEURS	GEOMETRIE	VISION	PRIMITIVES
EVENEMENTS	ETATS	ALEATOIRES	CHAINE	NIVEAU	SURFACES
TAUX	REPARTITION	CALCULS	PROPRES	GEOMETRIQUES	SEQUENCE
POLITIQUE	SYNCHRONE	LIMITE	DYNAMIQUE	CLASSIFICATION	COURBE
COMPOSITION	FINI	CITE	CONSIDERE	ORDONNANCEMENT	CORRESPONDANCE
SERVICES	ALEATOIRES	RESULTAT	PERMETTANT	TRAJECTOIRE	SCENE
CAPACITE	DETECTION	FILE	CALCULER	PERMETTANT	DETECTION

4.3 Représentation par SOM et IHM

4.3.1 Sur la représentation par carte de Kohonen

Représentation visuelle

Nous décrivons brièvement comment construire des représentations[78, 38, 79, 80, 81, 82, 83, 68, 84]. La métaphore du paysage fournit une vision locale des erreurs entre classes voisines et affiche le paysage calculé dans la matrice U[79], la matrice de distance entre classes voisines. La matrice U affiche la grille et pose une couleur entre deux centres, d'autant plus clair que ces centres sont proches dans l'espace des données. Ainsi, il se forme des vallées ou zones de la carte aux classes toutes voisines dans l'espace, séparées par des collines, ou zones frontière de la carte où les centres sont éloignés dans l'espace. Il est clair que le paysage obtenu doit être varié pour des données complexes, puisque sinon, on est en présence d'une surface non déformée (voisinages respectés parfaitement), ou d'une carte proche d'un tirage aléatoire (aucun voisinage respecté). Il est possible de naviguer dans les zones homogènes et de les caractériser par un label. Une classification (par exemple hiérarchique ou CAH) permet de séparer ces zones de façons automatiques, et faciliter leur localisation. Ces représentations illustrent l'indicateur global d'inertie intra étendue, le critère minimisé par SOM en analyse des données. Pour obtenir une vision plus globale de la déformation de la carte, on projette les centres à l'aide d'un critère globale comme[85] par une méthode MDS[86] par exemple ; de cette manière, l'allure générale de la carte apparaît. Des auteurs[68] projettent les centres sur une projection factorielle des données, en éléments supplémentaires ; l'échantillon doit être de taille raisonnable. De même, on peut superposer la projection des données, afin de vérifier visuellement la quantification finale. Ces représentations souffrent cependant de la déformation due au critère employé, et demeure indicative sur le respect de la topologie globale de la carte, ou relations de voisinage de la grille. En alternative, nous avons récemment proposé d'effectuer directement l'analyse factorielle sur les centres afin de dégager leurs corrélations ; les indicateurs de contribution, d'angle et de signe sont affichables pour chaque direction factorielle afin d'aider l'utilisateur dans son étude fine de la structure de la carte auto-organisée, et lui faciliter la navigation. Il a été également proposé diverses méthodes de projection des données classées autour de leur centre d'affectation. Par exemple, par des interpolations paramétrées à l'aide des classes voisines ou des classes dont les centres sont un peu moins proches que celle affectée au document à projeter. La projection factorielle est une alternative. Le plus simple reste de placer chaque document dans sa classe la plus proche, mais cette méthode ignore l'existence des moins proches voisins éventuellement éloignés sur la carte. Lorsqu'en plus de centres de classe pour le SOM, on dispose d'une distribution de classe comme pour le GTM, il devient possible de placer un document en sa moyenne conditionnelle de classe. Cette représentation a l'avantage de refléter la globalité de la cartographie au prix d'un risque d'erreur à l'interprétation des proximités pour des répartitions multimodales. Au lieu de représenter les défauts du voisinage qui indiquent les lieux de

la déformation de la carte, certains auteurs préfèrent projeter la densité du corpus, ou proportion de données classées dans chaque classe. Ils emploient éventuellement une interpolation sur les centres voisins afin d'accentuer les reliefs. Une telle représentation montre directement les groupes de documents similaires d'une part, et les documents plus spécifiques d'autre part. Il est clair que le choix de la représentation dépend de l'objectif de l'utilisateur. Des représentations simultanées doivent offrir un accès plus complet au contenu de la carte si la surcharge est évitée. Par exemple, on peut imaginer superposer au paysage de la matrice U les résultats de la projection factorielle en alternative à la classification seule. Dans la suite, nous superposons des cartes de densité bivariée du vocabulaire du corpus avec le paysage de la matrice U et de la densité projetée des données.

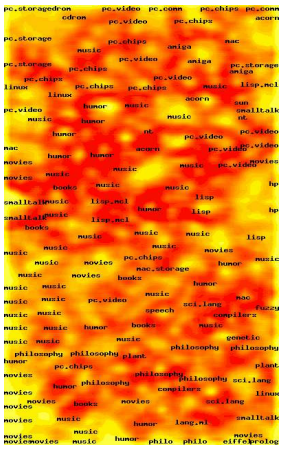
Problématique d'Interface Homme-Machine (IHM) et AD textuelles

Etant donné que l'espace de départ (données) est souvent de grande dimension, alors que celui d'arrivée (grille 2D) est de faible dimension, la projection par SOM induit une certaine déformation apparente, révélatrice de macro-clusters séparés par des frontières.

Les méthodes de représentation permettent de juger des vraies distances entre centres en référence avec la grille imaginaire. On a représenté ci-contre des exemples d'interface pour la navigation dans des archives de données textuelles (textes courts) ou sonores (morceaux MP3) ou d'image. On peut voir sur les trois premières vignettes, l'interface WEBSOM[87] avec matrice U sur la gauche, et une carte de densité [83] sur la droite pour des morceaux MP3 de musique. L'interface[47] du centre montre une représentation de résumés de film dans un espace non-euclidien. Les sorties suivantes correspondent sur la gauche à un SOM à hiérarchie pyramidale[88], au centre on trouve une représentation de la carte par classification hiérarchique[89] des centres, représentation proche des méthodes visuelles de *treemap*[90] et sur la droite un interface web pour interroger une base d'images.

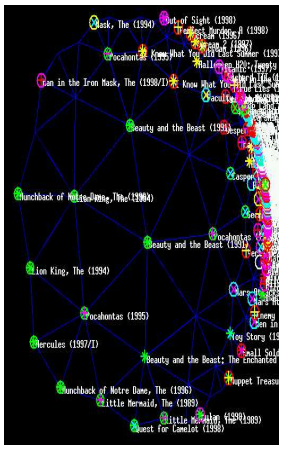
Dans le cas de données textuelles, la carte est étiquetée à l'aide des mots clefs pour chacune des classes de document afin de faciliter un accès visuel à de contenu. Un utilisateur peut alors consulter le détail du contenu en sélectionnant les zones qui l'intéressent. En outre, étant donné la taille des grilles parfois rencontrées, des méthodes de zoom sont proposées dans la littérature. Un zoom sur une zone particulière est possible, comme c'est le cas du système WEBSOM[87]. Il s'agit également d'une propriété du système de cartographique proposé récemment par un géographe ; il utilise un outil de navigation de carte routière pour afficher une carte de Kohonen. Il existe en outre des méthodes adaptées à l'affichage simultané de la carte globale (au moins aux alentour de la zone zoomée) et d'une zone accentuée à l'aide d'un effet visuel local. Les méthodes[91] de type *Polyfocal (Bifocal) Display* et *Fisheye Views*[92] effectuent des effets de zoom local, avec rétrécissement des zones frontières. Elles peuvent être complétées d'effets fractals[93] sur les zones alentours pour éventuellement en supprimer.

WEBSOM



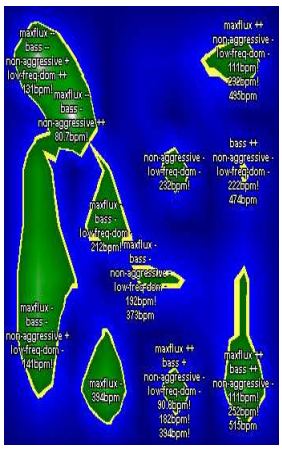
[Kohonen1997]

HSOM



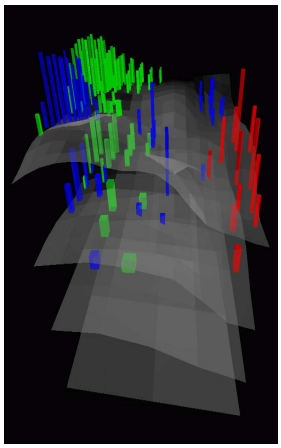
[Ritter1999]

Islands of Music



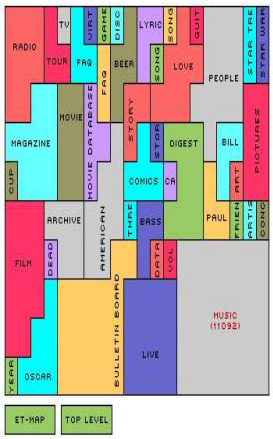
[Pampalk2001]

TS-SOM



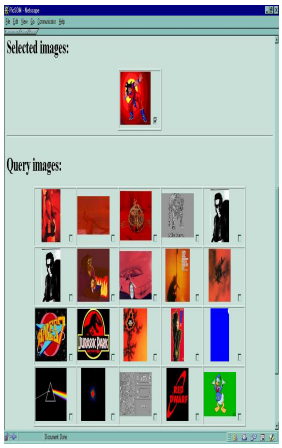
[Koikkalainen2002]

ET-Map



[Chen1998]

picSOM



[Laaksonen1999]

4.3.2 Prototype de recherche, IHM 3D

Il a été développé un prototype[1] C++ pour la projection par carte auto-organisatrice par l'algorithme CASOM, avec sortie textes, et extensions par scripts MATLAB. Une IHM rudimentaire pour la navigation dans une carte tridimensionnelle a été également implémentée en JAVA pour une meilleure lisibilité des résultats par l'utilisateur final. Il s'agit d'un prototype qui demanderait des recherches dans le domaine pour le finaliser.

[1]Nous tenons à remercier M. Yves LeChevallier, Directeur de recherche à l'INRIA, qui nous a très gentiment communiqué une première version d'un logiciel implémentant le SOM, et avec lequel le premier prototype a été conçu.

FIG. 4.18 – IHM 3D JAVA : 1

On peut trouver ci-contre, figure 4.18,4.19,4.20,4.21,4.22,4.23 un exemple de sortie de l'IHM JAVA. On reconnaît un paysage sur le graphique en couleur. Les pics sont des zones de fortes densités. On a représenté les probabilités a priori des classes multinomiales ou coefficients mélangeants estimés $\hat{\pi}_k$.

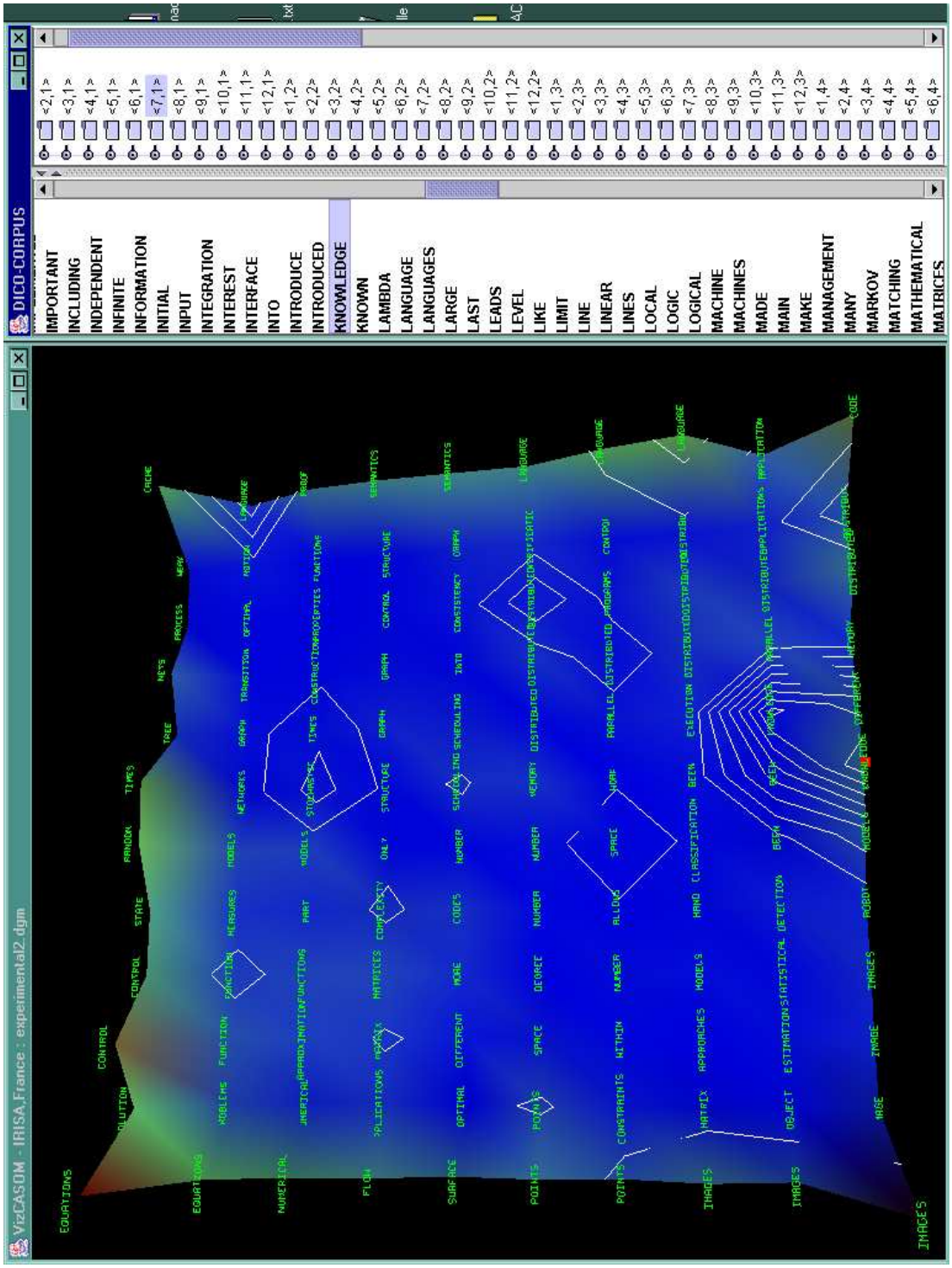


FIG. 4.19 – IHM 3D JAVA : 2

FIG. 4.20 – IHM 3D JAVA : 3

FIG. 4.21 – IHM 3D JAVA : 4

FIG. 4.22 – IHM 3D JAVA : 5

FIG. 4.23 – IHM 3D JAVA : 6

Chapitre 5

Conclusion et perspectives

5.1 Résultats et conclusions

Nous avons proposé dans ce document une description de diverses méthodes de projection des données qualitatives, notamment des lignes ou colonnes d'un tableau de contingence. L'illustration de ces méthodes sur des données textuelles connues pour leur complexité permet de tirer les avantages et les limites des méthodes proposées. La représentation bidimensionnelle offre un éventail de possibilités applicatives allant de la simple navigation dans un corpus à l'étude de contenu en passant par l'indexation des textes. Nous avons ainsi étudié et présenté tout particulièrement un point de vue vectoriel avec un formalisme probabiliste afin d'extraire de l'information contenue dans les textes. Le bilan de ces travaux est le suivant.

• Nous avons présenté une vue unifiée (et quasiment exhaustive) de la dizaine de méthodes probabilistes, à base de mélange de lois, permettant de construire des cartes auto-organisatrices, rassemblement bibliographique non réalisé jusqu'à présent à notre connaissance, et bien utile pour un lecteur intéressé par l'aspect modélisation des cartes de Kohonen.

• Nous avons ensuite proposé une nouvelle méthode, appelée **CASOM** que nous présentons comme une généralisation de l'Analyse Factorielle des Correspondances. En effet, elle projette les vecteurs de probabilité empirique à la manière du SOM (ou TPEM) et use d'une distance comparable à celle de l'AFC. Nous avons alors établi certaines propriétés originales de CASOM, parfois dérivant directement du SOM. Nous avons également avancé des hypothèses de travail capables de dégager des nouvelles propriétés statistiques des cartes auto-organisatrices. Il s'agit d'étudier les cartes du point de vue du χ^2, notamment en considérant le tableau de contingence sous-jacent aux multinomiales.

• Nous avons ensuite présenté un nouvelle façon d'aborder les modèles de carte en général. Il s'agit de s'inspirer du travail effectué en image pour modéliser les cartes de

Kohonen comme des modèles de Markov caché. Ce rapprochement a permis d'établir un nouvel algorithme de mélange flou pénalisé, le **TNEM**, qui nécessite après étude empirique une bonne initialisation. Autrement dit, si la version image est apte à procéder à de la segmentation contrainte, la version carte semble moins performante. En effet, les contraintes imposées par les cartes de Kohonen sont certainement moins informatives. Le TNEM pourrait donner des résultats intéressants en fin d'apprentissage de CASOM, si les probabilités a posteriori sont calculables. Vérifier que la plupart des algorithmes de segmentation d'image sont aisément modifiables en algorithmes de carte auto-organisatrice reste à faire.

• Nous avons également proposé un algorithme de projection nommé **CABSOM**, avec distance du χ^2, dont la complexité nous a semblé importante indépendamment du fait qu'elle procède en dimension K. Des extensions non linéaires sont a priori préférables pour éviter des inversion matricielles nombreuses, voire instables. Par contre, le choix de coefficients de classes floues au lieu de probabilités a posteriori, effectué au départ afin de parvenir à des modèles passant facilement à des échelles de données importantes nous est apparu finalement inintéressant du point de vue de l'estimation et de l'étude des propriétés résultantes. Des modèles pleinement probabilistes apparaissent plus séduisants du point de vue réglage du voisinage, point crucial non encore résolu pour CASOM, pour peu que de nouveaux modèles soient estimables pour des grandes bases de données. L'algorithme du CABSOM s'étend directement au cas pleinement probabiliste comme décrit dans la fin de la partie consacrée.

• Cette thèse s'est intéressée au formalisme des cartes de Kohonen en général et à celui pour tableaux de contingence en particulier. Elle a apporté modestement un complément théorique étant donné le formalisme particulier introduit ici. Elle aboutit à de nouveaux critères et permet de retrouver de plus connus. L'introduction de la notion du χ^2 est un complément à la notion généralement employée à propos des cartes auto-organisatrices : l'entropie de Shannon. L'utilisation des géostatistiques nous est apparue naturelle et pourrait constituer un axe de recherche des plus prometteurs pour la définition de critères de qualité mais également celle d'algorithmes robustes.

• Enfin ce travail a donné lieu au développement d'un prototype de recherche sous la forme de trois modules séparés, pour une projection des données textuelles. Le premier en langage C est chargé de construire très efficacement un dictionnaire à partir d'un fichier listant les textes du corpus. Un autre en langage C++ estime les cartes CASOM, en codant les matrices textuelles en une structure de données pour un format matriciel creux afin d'alléger la mémoire vive allouée et d'accélérer notablement les temps de calcul. Les sorties sont parfois traitées par des fonctions MATLAB. Enfin, une IHM prototype a été implantée sous JAVA 3D et VISAD, pour une navigation aisée dans le corpus projeté. Ces programmes, à la fin de la thèse, font l'objet d'un travail d'élèves (TER de maîtrise d'informatique) pour l'interface ou sont prévus pour une réécriture dans le langage JAVA (avec suppression des quelques scripts MATLAB

de post-traitement) qui est une bonne alternative pour des extensions futures.

5.2 Perspectives

5.2.1 Sur les représentations par carte

Le modèle proposé demeure difficile à estimer en pratique. En effet, dans la littérature, l'approche *batch* apparaît parfois comme moins robuste qu'une approche purement *stochastique*. Il serait intéressant d'étudier une estimation séquentielle. Il doit être également possible d'élaborer un algorithme automatique pour calculer une taille optimale du voisinage durant les itérations d'EM contraint. Ensuite, continuer sur l'idée de la comparaison avec l'image en approfondissant les modèles correspondants pourrait aboutir à un algorithme robuste de projection et surtout présentant des dépendances explicites entre classes. Un modèle complètement probabiliste pourrait présenter l'intérêt d'une meilleure robustesse, notamment en grandes dimensions.

Dans un autre ordre d'idées, la formalisation du voisinage est réalisable autrement que ce qui est connu aujourd'hui. On peut penser notamment à des modèles mécaniques par exemple. Encore faut-il qu'ils donnent de meilleurs résultats. En fait, le voisinage et sa manière de le représenter est sûrement la question fondamentale lors de la modélisation d'une carte. Les questions de taille, qualité de carte sont encore également des questions sans réponse. Des stratégies hiérarchiques déjà existantes dans le SOM originel apporteraient a priori un résultat plus efficace et plus robuste à CASOM, l'intérêt de CASOM par rapport au SOM, résidant dans son interprétabilité. Un test de taille de modèle pourrait s'effectuer entre deux transformations. Ensuite, plus particulièrement pour le domaine textuel, l'usage des multinomiales est licite puisqu'elles donnent en général des résultats parmi les meilleurs ; cependant, il serait intéressant de rechercher des distributions sans hypothèse d'indépendance entre les mots, et qui permettrait par la même occasion d'obtenir également les liaisons entre mots du vocabulaire.

On peut imaginer beaucoup de choses dans le domaine mais une modélisation vraiment proche du texte écrit semble la seule à même de véritablement comprendre un jour le contenu. La perte du flot du langage est manifestement irrémédiable, à moins de parler des langues mortes sans sémantique dans l'ordre des mots. C'est pourquoi, nous avons été amené à nous intéresser aux séquences de mots dans les phrases, par modélisation à l'aide des modèles graphiques (HMM). Un modèle mêlant les deux domaines carte-séquentiel permettrait des changements de point de vue, changements aptes à diminuer les erreurs d'interprétation dues au caractère artificiel de la modélisation.

5.2.2 Sur les représentations par graphe

La représentation planaire induit certainement une grande déformation de la structure des données. D'ailleurs, la notion même du codage vectoriel trouve ses limites dans l'élaboration d'un dictionnaire a priori figé et arbitraire. C'est pourquoi, nous avons été amené à nous intéresser à des modèles plus souples de représentation des textes, notamment les modèles de chaîne de Markov caché qui offrent la potentialité de représentation sous forme de graphe. Bien que cette idée n'ait guère été développée ici, l'énoncé d'un modèle particulier de texte a été proposé, étendant un existant particulièrement performant. Le développement de cette méthode devrait offrir des navigations dans les textes innovantes en permettant des représentations locales non déformées. D'ailleurs, l'usage des modèles bayésiens en texte est une réalité émergente et en pleine activité puisqu'ils permettent de prendre en compte non seulement la notion de séquence au niveau des mots mais également au niveau de la structure des documents. Or, justement, les documents multimédia sont structurés par construction, afin de pouvoir atteindre des contenus de formats variés (textes, son, image), et les pages en format HTML, SGML, XML ou autre suivent elles-mêmes une certaine structure, notamment arborescente.

Ainsi, nous avons pensé à étendre le PLSA au domaine séquentiel en considérant la dynamique des états de la variable latente dans un texte, sous la forme d'une chaîne de Markov cachée ou HMM[5, 4]. Il s'agit de l'idée d'un texte présentant une suite de concepts mis les uns à la suite des autres en un enchaînement précis ; pour simplifier, un concept dépendra uniquement de celui qui le précède :

Définition 36 *On définit le modèle du HMM-PLSA en considérant un corpus comme une suite de données dyadiques conditionnées par une variable inconnue discrète à valeurs dans $\mathcal{Z}$ et suivant une chaîne de Markov. On suppose observer I séquences x_i de longueur $N_{i\bullet} = |x_i|$ chacune. Elles sont considérées comme indépendantes :*

$$\mathcal{L}(\theta|\mathcal{D}) = \prod_{i\in\mathcal{I}} \sum_{\mathbf{z_i}\in\mathcal{Z}^{Ni\bullet}} P(z_{i1})P(x_{i1}|z_{i1})P(x_i|z_{i1}) \prod_{t=2}^{t=N_{i\bullet}} P(z_{it}|z_{i(t-1)})P(x_{it}|z_{it})P(x_i|z_{it}) \quad (5.1)$$

Le PLSA est clairement une simplification de ce modèle plus général. En effet, le PLSA considère l'indépendance des états de la variable cachée conjointement à l'événement observé (mot, document), alors qu'ici on cherche à estimer leur dépendance au premier ordre, au sens markovien du terme. On obtiendrait un graphe de dépendance donnant à la fois un descriptif condensé d'un corpus et un calcul efficace de la probabilité d'événement d'une séquence particulière de mots. Le modèle proposé partage certains points communs avec les modèles de langage, un label de document supplémentaire. Pour estimer les paramètres, on peut maximiser par exemple la vraisemblance et utiliser un algorithme de type EM associé aux algorithmes Backward-Forward qui permettent efficacement de calculer les probabilités de mise à jour de l'EM. Nous ne développons pas les méthodes de Viterbi et Backard-Forward qui sont décrites dans l'article de

référence [5]. D'un point de vue applicatif, un utilisateur passe d'un contexte à l'autre dans le graphe à l'aide des probabilités de transition, à partir d'une requête initialisant la recherche. Une manière d'accélérer le processus consiste à se placer sur les cartes factorielles de la matrice de transition afin de cibler les concepts clefs auxquels s'intéresse l'utilisateur du système. Nos résultats préliminaires basés sur un K-means segmental n'ont pas donnés les résultats escomptés. Cependant, on peut retrouver un modèle très performant et proche, appelé CHMM introduit dans [94] et dont l'auteur se sert pour segmenter des chaînes d'ADN. Le CHMM travaille sur des labels de classe au lieu de labels de documents. Nous avons présenté ce modèle en raison de son intérêt pratique en analyse textuelle : une représentation sous forme de graphe d'un corpus et les utilisations directement exploitables de formules de mises à jour des probabilités dans les HMM qui pourraient permettre de travailler sur des bases de données dynamiques. Des critères de recherche d'information comme dans [95] sont également envisageables. Ce modèle assez basique doit pouvoir s'étendre à des modèles plus robustes prenant en compte plutôt qu'un mot, une fenêtre glissante de mots afin de robustifier les estimations et les rendre moins sensibles au style des textes. Les modèles de classification récents à base d'HMM montrent également que ce genre d'approche doit donner des résultats satisfaisants. Il se trouve qu'un modèle[96] très similaire a été proposé récemment et indépendamment.

Annexe A

A.1 Exemple de lois discrètes

Il est possible d'écrire les loi suivantes ainsi que bon nombre d'autres telle que gaussienne sous une forme paramétrique générique : la famille des loi exponentielles. Ces lois ont de bonnes propriétés en général.

A.1.1 Modèle multi-varié de Bernoulli

On suppose que l'événement associé à la présence ou absence d'un mot (quelque soit son occurrence) conditionnellement à l'événement $z \in \mathcal{Z}$ est une variable de Bernoulli B_{ik}, qui prend la valeur 1 si le mot v_j est présent dans x_i, et 0 sinon ; soit :

$$P(x_i|z,\theta) = \prod_{j=1}^{j=J} P(v_j|z,\theta) = \prod_{j=1}^{j=J} {P_{j|z}}^{B_{ij}} (1 - P_{j|z})^{1-B_{ij}} \tag{A.1}$$

On modélise la probabilité d'apparition d'un document comme l'ensemble des événements indépendants de présence ou absence d'un mot, ici considéré comme événement issu d'un tirage bernoullien.

A.1.2 Modèle multivarié de Poisson

On utilise cette distribution pour des événements rares. Il s'agit pour chaque composante de lois limites bernoulliennes :

$$P(x_i|z,\theta) = \prod_{j=1}^{j=J} P(v_j|z,\theta) = \prod_{j=1}^{j=J} \exp(\lambda_{j|z}) \frac{\lambda_{j|z}^{N_{ij}}}{N_{ij}!} \tag{A.2}$$

A.1.3 Modèle multinomial

L'hypothèse '*Bayes Naïf*' est la suivante : ni la place, ni l'ordre des mots dans un texte ne sont porteurs d'information, il y a indépendance pour chacun des mots du texte vis à vis des mots le précédant conditionnellement à l'événement E. Ainsi, si x_i est la

suite de mots $x_{i1}, x_{i2}, \cdots, x_{i|x_i|}$, pour $|x_i| = N_{i\bullet}$, on note N_{ij}[1] le nombre d'occurrence du mot v_j dans le texte x_i :

$$P(x_i|z,\theta) = \prod_{t=1}^{t=|x_i|} P(x_{it}|z,\theta, x_{iu} \forall u < t) = \prod_{j=1}^{j=J} {P_{j|z}}^{N_{ij}} \tag{A.3}$$

Soit, à un facteur multiplicatif près, cela est équivalent à prendre en compte explicitement la longueur des documents, en considérant le tirage des mots comme multinomiaux avec autant de tirages indépendants que de mots ; soit :

$$P(x_i|z,,\theta) = N_{i\bullet}! \prod_{j=1}^{j=J} (\frac{P_{j|z}^{N_{ij}}}{N_{ij}!}) \tag{A.4}$$

Cette loi permet de construire des séparations linéaires entre classes.

A.2 Loi de $E_{\mathcal{CASOM}}(\hat{\theta}, \hat{\mu}|\mathcal{D})$

Pour rappel, on énonce la propriété à montrer.

Propriété 8 *Si on note :*

$$\begin{array}{ll} \bullet\ \hat{m}_{\mathcal{CASOM}} & = \sum_k (\sum_i \hat{\mu}_{ik} f_i) \sum_{l \neq k} h_{kl} KL(\hat{P}_{k\bullet}||\hat{P}_{l\bullet}) \approx \sum_k \hat{\pi}_k \sum_{l \neq k} h_{kl} KL(\hat{P}_{k\bullet}||\hat{P}_{l\bullet}) \\ \bullet\ \hat{a}_{jk} & = h_{k\bullet}(1 + \log \hat{P}_{j|k}) - \log \prod_l \hat{P}_{j|l}^{h_{kl}} \approx \sum_l h_{kl} \log(\hat{P}_{j|k}/\hat{P}_{j|l}) \\ \bullet\ \hat{\sigma}_{\mathcal{CASOM}} & = \sqrt{\sum_i f_i [\sum_j \hat{P}_{j|\hat{z}_i} \hat{a}^2_{j\hat{z}_i} - \sum_{j_1} \sum_{j_2} \hat{P}_{j_1|\hat{z}_i} \hat{P}_{j_2|\hat{z}_i} \hat{a}_{j_1\hat{z}_i} \hat{a}_{j_2\hat{z}_i}]} \end{array} \tag{A.5}$$

Alors, pour $min_i N_{i\bullet}$ assez grand, et sous l'hypothèse que $\hat{\theta}$ est le vrai paramètre, on a le résultat de convergence asymptotique en loi :

$$\frac{E_{\mathcal{CASOM}}(\hat{\theta}, \hat{\mu}|\mathcal{D}) - \hat{m}_{\mathcal{CASOM}}}{\hat{\sigma}_{\mathcal{CASOM}}} \sim \mathcal{N}_1(0,1) \tag{A.6}$$

□

Pour la démonstration de cette propriété, nous proposons le lemme suivant :

Lemme A.2.1
Soit un ensemble de données i.i.d. x_i, pour i de 1 à I, et vérifiant les hypothèses de classe suivantes :

$$\forall i\ \exists!\ k\ \ tq.\ x_i \sim X^k,\ \ où\ X^k\ v.a.\ tq.\ \mathbb{E}(X^k) = E_k\ et\ \mathbb{V}(X^k) = \Sigma_k, \tag{A.7}$$

On note également :

[1] Il s'agit de x_{ij} pour rester homogène avec les définitions précédentes

1. *les variables binaires d'attribution des données à leur classe* c_{ik},
2. *les moyennes empiriques de classes* $\bar{x}^k = \frac{\sum_i c_{ik} x_i}{\sum_i c_{ik}}$,
3. *le nombre de réalisations par classe* $n_k = \sum_i c_{ik}$,
4. *les fonctions* g_k *de* $\mathcal{R}$ *dans* $\mathcal{R}$, *et telles que* $\phi_k = \nabla g_k(x)|_{x=E_k}$,
5. *une transformation linéaire par les* $a_k \geq 0$,

Alors, d'après les hypothèse de classe, on a :

$$\frac{\sum_k a_k \sqrt{n_k}(g_k(\bar{x}^k) - g_k(E_k))}{\sqrt{\sum_k a_k^2 \phi_k^T \Sigma_k \phi_k}} \xrightarrow{\mathcal{L}} \mathcal{N}_1(0,1) \tag{A.8}$$

Preuve (du lemme)
En effet on a clairement sous les hypothèses :

$$\begin{array}{ll}
^{(*)} & \forall k \ \sqrt{n_k}(\bar{x^k} - E_k) \xrightarrow{\mathcal{L}} \mathcal{N}_J(0, \Sigma_k) \\
\overset{(**)}{\Rightarrow} & \forall k \ \sqrt{n_k}(g_k(\bar{x^k}) - g_k(E_k)) \xrightarrow{\mathcal{L}} \mathcal{N}_1(0, \phi_k^T \Sigma_k \phi_k) \\
\overset{(***)}{\Rightarrow} & \begin{pmatrix} \sqrt{n_1}(g_1(\bar{x^1}) - g_1(E_1)) \\ \sqrt{n_2}(g_2(\bar{x^2}) - g_2(E_2)) \\ \vdots \\ \sqrt{n_K}(g_K(\bar{x^K}) - g_K(E_K)) \end{pmatrix} \xrightarrow{\mathcal{L}} \mathcal{N}_K \begin{pmatrix} \phi_1^T \Sigma_1 \phi_1 & & & \\ & \phi_2^T \Sigma_2 \phi_2^T & (0) & \\ & (0) & \ddots & \\ & & & \phi_K^T \Sigma_K \phi_K^T \end{pmatrix}
\end{array} \tag{A.9}$$

En multipliant par le vecteur $(a_1, a_2, \cdots, a_k)^T$, on obtient l'énoncé du lemme. De plus, on a noté (*) pour *par le TCL*, (**) pour *par la règle Delta* et (***) pour *par indépendance.* □

On en déduit la preuve du théorème précédent :

Preuve (de la propriété)
On fait l'hypothèse forte d'identification des vecteurs multinomiaux estimés avec les vrais ($\hat{\theta} = \theta^v$), hypothèse de test raisonnable pour un assez grand nombre de documents. On applique alors le lemme précédent sur les moyennes $f_{j|i}$, vues comme des moyennes de Bernoulli, et le critère comme une somme de fonctions avec le vecteur de coefficients multiplicateurs $a = (\sqrt{f_{1\bullet}}, \sqrt{f_{2\bullet}}, \cdots, \sqrt{f_{I\bullet}})^T$. Il s'agit d'un vecteur non constant (supposé convergent) ; c'est pourquoi, on justifie finalement la convergence en loi grâce au théorème de Slustsky. □

Bibliographie

[1] C. J. Van Rijsbergen, *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[2] G. Salton and M. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., 1983.

[3] L. Lebart and A. Salem, "Statistique textuelle," *Dunod*, 1994.

[4] F. Jelinek, *Statistical methods for speech recognition.* The MIT Press, 1997.

[5] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," 1989.

[6] H. Luhn, "A statistical approach to mechanised encoding and searching of library information," *IBM Journal of Research and Development 1 : 309-317*, 1957.

[7] D. A. Grossman and O. Frieder, "Information retrieval - algorithm and heuristics," *Kluwer Academic Publishers*, 1998.

[8] M. Sahami, "Learning limited dependence bayesian classifiers," in *Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 335–338, CA :AAAI Press, 1996.

[9] R. Fung and B. D. Favero, "Applying bayesian networks to information retrieval," *ACM*, vol. 38, no. 3, 1995.

[10] H. Turtle and W. B. Croft, "Inference netwoks for documents retrieval," *ACM*, 1990.

[11] M. E. Frisse and S. B. Cousins, "Information retrieval from hypertext : Update on the dynamic handbook project," *Hypertext'89*, 1989.

[12] J. Savoy and D. Desbois, "Réseaux d'inférence bayésiens dans un système hypertexte : Principe et premiers résultats," *Université de Montréal*, 1990.

[13] L. Lebart, A. Morineau, and M. Piron, *Statistique exploratoire multidimensionnelle.* Dunod, 1997.

[14] L. Kaufman and P. J. Rousseuw, *Finding Groups in Data - An introduction to cluster Analysis.* Wiley Series in Probability and Mathematial Statistics, 1990.

[15] G. Saporta, *Probabilités, Analyse de Données et Statistique.* Edition Technip, 1990.

[16] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Stat. and Proba.*, vol. 1, pp. 281–296, 1967.

[17] S. Lloy, "Least squares quantization in pcm," *IEEE Trans. Information Theory*, vol. 28, pp. 129–137, 1982.

[18] M. Duflo, *Algorithmes stochastiques 23.* Mathématiques et Applications, Springer, 1996.

[19] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceeding of The IEEE, Vol. 86, N11, November 1998, pp 2210-2239*, 1998.

[20] S. Kirkpatrick, C. Gelatt, and M.P.Vecchi, "Optimisation by simulated annealing," *Science*, no. 220, 1983.

[21] A. McCallum and K. Nigam, "A comparaison of event models for naives bayes text classification," in *AAAI-98 Workshop on Learning Text Categorization*, 1998.

[22] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the em algorithm," *J. Royal Statist. Soc. Ser. B., 39*, 1977.

[23] J.-J. Droesbeke, B. Fichet, and P. Tassi, *Modèles pour l'analyse des données multidimensionnelles.* Economica, 1992.

[24] S. Russell, "The EM algorithm." CS 281 Machine Learning Spring 1998.

[25] R. Redner and H. Walker, "Mixture densities, maximum likelihood, and the EM algorithm," *SIAM*, vol. 26, pp. 195–239, 1984.

[26] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for gaussian mixtures," *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996.

[27] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastics versions," *Computational Statistics and Data Analysis*, no. 14, pp. 315–332, 1992.

[28] R. Hathaway, "Another interpretation of the EM algorithm and extensions," *Statistics & Probability Lettes*, vol. 4, pp. 53–56, 1986.

[29] T. P. Minka, "Automatic choice of dimensionality for pca," *NIPS*, vol. 13, 2000.

[30] T. Hoffman, "Probabilistic latent semantic analysis," *SIGIR'99*, 1999.

[31] T. Hofmann, "Learning probabilistic models of the web," in *Research and Development in Information Retrieval*, pp. 369–371, 2000.

[32] D. A. Harville, *Matrix algebra from a statistician's perspective.* Springer, 1997.

[33] J. P. Benzecri, *L'analyse des données tome 1 et 2 : l'analyse des correspondances.* Paris :Dunod, 1980.

[34] P. Michel, *Statistique Exploratoire Multivariée.* ENSAI, 1999.

[35] S. Deerwester, S. T. Dumais, G. W. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science 41(6) :391-407*, 1990.

[36] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, pp. 502–516, 1989.

[37] K.-Y. Chang and J. Ghosh, "A unified model for probabilistic principal surfaces," *IEEE Transaction on Pattern Analysis and Machine Learning*, vol. 23, no. 1, pp. 22–40, 2001.

[38] T. Kohonen, *Self-organizing maps*. Springer, 1997.

[39] T. Kohonen, "The Self-organizing Map (SOM)," tech. rep., Helsinki University of Technology, 1999.

[40] F. Mulier and V. Cherkassky, "Self-organization as an iterative kernel smoothing process," *Neural Computation*, 1995.

[41] M. Cottrell, J. Fort, and G. Pagès, "Theoretical aspects of the SOM algorithm," *Neurocomputing 21 (1998) 119-138*, 1998.

[42] H. Yin and N. Allinson, "On the distribution and convergence of feature space in self-organizing maps," *Neural Computation*, no. 7, pp. 1178–1187, 1995.

[43] E. Erwin, K. Obermayer, and K. Shulten, "Self-organizing maps : ordering, convergence properties and energy functions," *Biol. Cyb.*, no. 67, pp. 47–55, 1992.

[44] T. Kohonen, "Self-organizing maps : Optimization approaches," *Artificial Neural Networks*, 1991.

[45] B. Fritzke, "Some competitive learning methods." Systems Biophysics, Institute for Neural Computation, Ruhr-Universitat Bochum, april 1997.

[46] M. Dittenbach, A. Rauber, and Merkl, "Recent advances with the growing hierarchical self-organizing map," in *3rd Workshop on Self-Organising Maps*, pp. 140–145, Springer-Verlag, 2001.

[47] J. Ontrup and H. Ritter, "Hyperbolic self-organizing maps for semantic navigation," *Advances in Neural Information Processing Systems 14*, 2001.

[48] B. Meyer, "Self-organizing graphs — A neural network perspective of graph layout," *Lecture Notes in Computer Science*, vol. 1547, pp. 246–262, 1998.

[49] C. M. Bishop, M. Svensén, and C. K. I. Williams, "Gtm : A principles alternative to self-organizing map," *Advances in Neuronal Processing System 9, MIT Press*, 1997.

[50] M. E. Tipping and C. M. Bishop, "Mixture of probabilistic principal component analysers," *Neural Computation*, vol. 2, no. 11, 1999.

[51] S. Roweis, "EM algorithms for PCA and SPCA," in *Advances in Neural Information Processing Systems* (M. I. Jordan, M. J. Kearns, and S. A. Solla, eds.), vol. 10, The MIT Press, 1998.

[52] C. M. Bishop, M. Svensén, and C. K. I. Williams, "Developpements of generative topographic mapping," *Neurocomputing*, vol. 21, pp. 203–224, 1998.

[53] C. M. Bishop, M. Svensén, and C. K. I. Williams, "GTM : The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215–234, 1998.

[54] M. Girolami, "Document representation based on generative multivariate bernoulli latent topics models," in *BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research* (U. of Cambridge, ed.), 2001.

[55] M. E. Tipping, "Probabilistic visualisation of high-dimensionnal binary data," *Advances in Neural Information Processing Systems*, no. 11, pp. 592–598, 1999.

[56] R. Tibshirani, "Principal curves revisited," *Statistics and Computing*, vol. 2, pp. 183–190, 1992.

[57] A. Utsugi, "Topology selection for self-organizing maps," *Network-Computation in Neural Systems*, vol. 7, no. 4, pp. 727–740, 1996.

[58] A. Utsugi, "Hyperparameter selection for self-organizing maps," *Neural Computation, 9 :623–635*, 1997.

[59] A. Utsugi, "Density estimation by mixture models with smoothing priors," *Neural Computation*, vol. 10, no. 8, pp. 2115–2135, 1998.

[60] A. Kaban and M. Girolami, "A combined latent class and trait model for analysis and visualisation of discrete data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[61] C. Ambroise and G. Govaert, "Constrained clustering and kohonen self-organizing maps," *Journal of Classification*, vol. 13, no. 2, pp. 299–313, 1996.

[62] S. Luttrell, "A bayesian analysis of self-organizing maps," *Neural Computation*, vol. 6, pp. 767–794, 1994.

[63] F. Anouar, F. Badran, and S. Thiria, *Cartes Topologiques et Nuées Dynamiques*, ch. 11. Thiria et al-Dunod, 1997.

[64] T. Hofmann, "Probabilistic topic maps : Navigating throught large text collections," *IDA'99, LNCS 1642, pp 161-172*, 1999.

[65] T. P. Minka, "Bayesian inference of a multinomial distribution." tutorial, july 2001.

[66] A. Agresti, *Categorial Data Analysis*. Wiley Series in probability and mathematical statitics, 1990.

[67] G. Celeux and G. Govaert, "Clustering criteria for discrete data and latent class models," *Journal of Classification*, vol. 8, pp. 157–176, 1991.

[68] O. Elemento, "Initialisation, convergence, et validation de cartes topologiques de kohonen," Master's thesis, stage INRIA (Yves Lechevallier), 1999.

[69] E. de Bodt, M. Cottrell, and M. Verleysen, "Are they really neighbor ? a statistical analysis of the SOM algorithm output.," *SAMOS preprint*, 2000.

[70] G. Parisi, *Statistical Field Theory*. Addison-Wesley, 1988.

[71] J. Zhang, "The mean field theory in EM procedures for markov random fields," *IEEE Transactions on Signal Processing*, vol. 10, no. 40, pp. 2570–2583, 1992.

[72] M. V. Dang, *Classification de données spatiales : modèles probabilistes et critères de partitionnement*. PhD thesis, Université de Technologie de Compiègne, 1998.

[73] N. Peyrard, *Approximations de type champ moyen des modèles de champ de Markov pour la segmentation de données spatiales*. PhD thesis, INRIA Rhône-Alpes, 2001.

[74] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the bayesian restoration of images," *IEEE Trans. Pattn. Anal. Mach. Intell.*, vol. 6, pp. 721–741, 1984.

[75] J. Wilkinson, *The Algebraic Eigenvalue Problem.* Clarendon Press, 1965.

[76] A. Morin, M. Kerbaol, and J. Bansard, "Etude des résumés en français des rapports de recherche d'un institut d'informatique publiés de 1989 à 1998," in *JADT'2000*, 2000.

[77] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization* (A. Press, ed.), pp. 41–48, 1998.

[78] J. Vesanto, "Using SOM in data mining, licenciate's thesis of technology," 2000.

[79] A. Ultsch, "New approaches in classification and data analysis. Integration of neural networks with symbolic knowledge processing," *Springer Verlag*, pp. 445–454, 1994.

[80] A. Ultsch and C. Vetter, "Self-organizing feature maps versus statistical clustering : A benchmark," *Research Report No. 9*, 1994.

[81] S. Kaski, "Data exploration using self-organizing maps," *Acta Polytechnica Scandinavica, Mathematics, Computing and Management*, March 1997.

[82] J. Vesanto, "SOM-based data visualization methods," *Intelligent-Data-Analysis*, no. 3, pp. 111–26, 1999.

[83] E. Pampalk, A. Rauber, and D. Merkl, "Using smoothed data histograms for cluster visualization in self-organizing maps," in *Proceedings of the International Conference on Artifical Neural Networks (ICANN'02)*, (Madrid, Spain), Springer, August 27-30 2002.

[84] C. Ambroise, *Approche probabiliste en classification automatique et contraintes de voisinage.* PhD thesis, UTC, 1996.

[85] J. Vesanto, J. Himberg, M. Siponen, and O. Simula, "Enhancing SOM based data visualization," in *Proceedings of the International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA'98)*, (Iizuka, Japan), pp. 64–67, October 1998.

[86] J. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 5, pp. 401–409, may 1969.

[87] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, and V. P. et A. Saarela, "Self organization of a massive document collection," *IEEE Transactions on Neural Networks*, vol. 11, pp. 574–585, 2000.

[88] A. Lensu and P. Koikkalainen, "A parallel implementation of the tree-structured self-organizing map," in *6th International Conference on Applied Parallel Computing*, pp. 370–379, 2002.

[89] J. Vesanto and E. Alhonieni, "Clustering of the self-organizing map," *IEEE Neural Networks*, vol. 3, no. 11, 2000.

[90] B. Bederson, B. Shneiderman, and M. Wattenberg, "Ordered and quantum treemaps : Making effective use of 2d space to display hierarchies," *ACM Transactions on Graphics (TOG)*, vol. 4, no. 21, pp. 833–854, October 2002.

[91] Y. Leung and M. Apperley, "A review and taxonomy of distortion-oriented presentation techniques," in *Readings in Information Visualisation : Using Vision to Think* (S.Card, J. Mackinlay, and B.Shneiderman, eds.), Morgan Kaufmann, 1999.

[92] G. W. Furnas, "Generalized fisheye views," in *Human Factors in Computing Systems (CHI '86)*, pp. 16–23, April 1986.

[93] H. Koike, "Fractal views : a fractal-based method for controlling information display," *ACM Transactions on Information Systems (TOIS)*, vol. 13, no. 3, pp. 305–323, 1995.

[94] Krogh, "Hidden markov models for labeled sequences," *ICPR'94*, vol. 12, pp. 140–144, 1994.

[95] K. Ng, "A maximum likelihood ratio information retrieval model," in *Eighth Text Retrieval Conference (TREC-8)*, 1999.

[96] D. M. Blei and P. J. Moreno, "Topic segmentation with an aspect hidden markov model," Tech. Rep. CRL 2001/07, Cambridge Research Laboratory, July 2001.

Table des matières

Table des figures

Résumé : Cette thèse s'intéresse à l'analyse exploratoire des données multi-dimensionnelles souvent qualitatives voir textuelles par des modèles particuliers de carte auto-organisatrice de Kohonen. Il s'agit d'effectuer une classification et une projection simultanées des lignes ou colonnes d'une matrice de données. Le résultat de ces méthodes est une réduction sous la forme d'une surface de régression discréte. Nous étudions plus particulièrement les modèles de mélange de lois de probabilités : les paramètres correspondant aux espérances des vecteurs classés sont contraints en les plaçant aux noeuds d'une grille rectangulaire. Après une présentation de ces méthodes, et des algorithmes d'estimation basés sur l'EM(*Expectation-Maximization*), nous introduisons essentiellement deux nouvelles approches. La première vise à "*généraliser la méthode d'Analyse Factorielle des Correspondances*" aux grandes matrices : l'algorithme CASOM est une classifieur Naïf de Bayes contraint en un TPEM(*Topology Preserving EM*) pour tableau de contingence. La seconde consiste en un schéma général d'adaptation des méthodes de segmentation d'image en carte auto-organisatrice. Pour l'illustrer, nous modifions un algorithme de segmentation par champs moyens, et obtenons un algorithme appellé TNEM. Nous utilisons ces méthodes pour aider à la navigation dans un corpus textuel. En effet, nous aboutissons à des critères et des moyens de représentation objectifs.

Mots-clefs : Cartes auto-organisatrices, mélange de lois de probabilité contraintes, analyse exploratoire des tableaux de contingences.

Summary : *This thesis is concerned with exploratory analysis of multidimensionnel data which are often qualitative to see textual, by particular Kohonen's self-organizing map models. The goal is to cluster and project simultaneously lines or columns of a data matrix. The result of these methods is a reduction in the form of a discrete surface of regression. We study most precisely mixture models of probabilistic laws : the parameters corresponding to means of clustered vecteurs are constrained by setting them at the nodes of a rectangular mesh. After an overview of theses methods, and the learning algorithms based on EM (Expectation-Maximization), we introduce two new approaches. The first one aims at "generalizing the Correspondence Analysis" method to big matrices : the CASOM algorithm is a Naive Bayes classifier which is constrained as a TPEM (Topology Preserving EM) for a contingency table. The second one consists of mutating image clustering algorithms into map algorithms. As an illustration, we modify a clustering algorithm based on mean-field, and we get an algorithm that we name TNEM. We use these methods to help with navigation in a textual corpus. Indeed, we end to objective criteria and cartographies.*

Key-words : *Self-organizing maps, mixture of probabilistic constrained laws, exploratory analysis of contingency tables.*